



Docket No.: 20212-515

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of : Customer Number: 20277  
Tetsuo MORITA : Confirmation Number: 1625  
Serial No.: 10/616,304 : Group Art Unit: 2171  
Filed: July 10, 2003 : Examiner:  
For: TRANSMISSION DATA GENERATION METHOD AND TRANSMISSION  
DATA GENERATION EQUIPMENT

**TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT(S)**

Mail Stop Missing Parts  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

At the time the above application was filed, priority was claimed based on the following application:

**Japanese Patent Application No. 2003-132292, filed May 9, 2003**

A copy of each priority application listed above is enclosed.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

Arthur J. Steiner  
Registration No. 26,106

600 13<sup>th</sup> Street, N.W.  
Washington, DC 20005-3096  
(202) 756-8000 AJS:prg  
Facsimile: (202) 756-8087  
**Date: December 10, 2003**



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

50212-515  
Morita  
July 10, 2003  
10/616,304

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 3 年 5 月 9 日

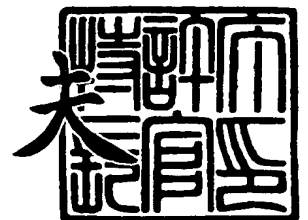
出 願 番 号  
Application Number: 特 願 2 0 0 3 - 1 3 2 2 9 2  
[ST. 10/C]: [ J P 2 0 0 3 - 1 3 2 2 9 2 ]

出 願 人  
Applicant(s): 住友電気工業株式会社

2 0 0 3 年 8 月 2 9 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 7 0 7 4 9

【書類名】 特許願

【整理番号】 102Y0726

【提出日】 平成15年 5月 9日

【あて先】 特許庁長官殿

【国際特許分類】 H03M 7/00  
G06F 15/00

【発明者】

【住所又は居所】 大阪府大阪市此花区島屋一丁目 1 番 3 号 住友電気工業  
株式会社大阪製作所内

【氏名】 森田 哲郎

【特許出願人】

【識別番号】 000002130

【氏名又は名称】 住友電気工業株式会社

【代理人】

【識別番号】 100088155

【弁理士】

【氏名又は名称】 長谷川 芳樹

【選任した代理人】

【識別番号】 100089978

【弁理士】

【氏名又は名称】 塩田 辰也

【選任した代理人】

【識別番号】 100092657

【弁理士】

【氏名又は名称】 寺崎 史朗

【選任した代理人】

【識別番号】 100110582

【弁理士】

【氏名又は名称】 柴田 昌聰



【パリ条約による優先権等の主張】

【国名】 アメリカ合衆国

【出願日】 2002年 7月12日

【出願番号】 60/395331

【手数料の表示】

【予納台帳番号】 014708

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0106993

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 伝送データ生成方法及び伝送データ生成装置

【特許請求の範囲】

【請求項1】 固定ブロックのサイズをオーバーヘッドに基づいて設定する固定ブロックサイズ設定工程と、

コンテンツのセグメント毎に、前記固定ブロックのサイズに基づいてセグメントのサイズを算出するセグメントサイズ算出工程と、

前記算出したセグメントのサイズに応じてコンテンツをセグメントに分割するセグメント分割工程と、

前記分割したセグメントをブロックに分割するブロック分割工程と、

前記分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ化工程と

を含み、

前記固定ブロックにおけるオーバーヘッドに基づいてコンテンツのセグメント毎にオーバーヘッドが設定されることを特徴とする伝送データ生成方法。

【請求項2】 固定ブロックのサイズをオーバーヘッドに基づいて設定する固定ブロックサイズ設定工程と、

前記固定ブロックのサイズに基づいて前記固定ブロックの再生時間を算出する固定ブロック再生時間算出工程と、

コンテンツのセグメント毎に、前記固定ブロックの再生時間に基づいてセグメントの再生時間を算出する再生時間算出工程と、

コンテンツのセグメント毎に、算出済みのセグメントの再生時間に基づいてセグメントの伝送時間を算出する伝送時間算出工程と、

前記算出したセグメントの伝送時間に応じてコンテンツをセグメントに分割するセグメント分割工程と、

前記分割したセグメントをブロックに分割するブロック分割工程と、

前記分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ化工程と

を含み、

前記固定ブロックにおけるオーバーヘッドに基づいてコンテンツのセグメント毎にオーバーヘッドが設定されることを特徴とする伝送データ生成方法。

【請求項 3】 前記固定ブロックサイズ設定工程では、オーバーヘッドが小さい値となるように前記固定ブロックのサイズを設定することを特徴とする請求項 1 又は請求項 2 に記載する伝送データ生成方法。

【請求項 4】 セグメントのサイズが固定ブロックのサイズの整数倍でない場合、コンテンツのセグメント毎に、前記固定ブロックで分割できない任意ブロックのサイズ又は再生時間と任意ブロックにおけるオーバーヘッドとを求める任意ブロック設定工程を含み、

セグメントのサイズが固定ブロックのサイズの整数倍でない場合、前記セグメント毎のオーバーヘッドが、前記固定ブロックにおけるオーバーヘッド及び前記セグメントの任意ブロックにおけるオーバーヘッドに基づいて設定されることを特徴とする請求項 1 ～請求項 3 のいずれか 1 項に記載する伝送データ生成方法。

【請求項 5】 前記固定ブロックの再生時間に基づいて前記固定ブロックの伝送時間を算出する固定ブロック伝送時間算出工程と、

コンテンツのセグメント毎に、前記セグメントの伝送時間及び前記固定ブロックの伝送時間に基づいてセグメントに含まれる固定ブロックの数を算出する固定ブロック数算出工程と、

コンテンツのセグメント毎に、前記算出した固定ブロックの数及び前記固定ブロックの再生時間に基づいてセグメントに含まれる全ての固定ブロックの再生時間を算出する全固定ブロック再生時間算出工程と

を含み、

前記再生時間算出工程では、コンテンツのセグメント毎に、セグメントのサイズが固定ブロックのサイズの整数倍の場合には前記セグメントに含まれる全ての固定ブロックの再生時間をセグメントの再生時間とし、セグメントのサイズが固定ブロックのサイズの整数倍でない場合には前記セグメントの任意ブロックの再生時間及び前記セグメントに含まれる全ての固定ブロックの再生時間に基づいてセグメントの再生時間を算出することを特徴とする請求項 2 ～ 4 のいずれか 1 項に記載する伝送データ生成方法。

【請求項 6】 前記任意ブロック設定工程では、コンテンツのセグメント毎に、前記セグメントに含まれる全ての固定ブロックの再生時間及び前記セグメントの伝送時間を用いて前記任意ブロックの再生時間と前記任意ブロックにおけるオーバーヘッドとの乗算値を求め、当該乗算値から所定の数値解析法により前記任意ブロックの再生時間及び前記任意ブロックにおけるオーバーヘッドを求めることを特徴とする請求項 5 に記載する伝送データ生成方法。

【請求項 7】 固定ブロックのサイズをオーバーヘッドに基づいて設定し、当該固定ブロックのサイズに基づいて固定ブロックの再生時間を算出し、コンテンツのセグメント毎に、当該固定ブロックの再生時間に基づいてセグメントの再生時間を算出し、算出済みのセグメントの再生時間に基づいてセグメントの伝送時間を算出する時間算出手段と、

前記時間算出手段で算出したセグメントの伝送時間に応じてコンテンツをセグメントに分割し、当該分割したセグメントをブロックに分割する分割手段と、

前記分割手段で分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ手段と

を備え、

前記時間算出手段は、当該固定ブロックにおけるオーバーヘッドに基づいてセグメント毎にオーバーヘッドを設定することを特徴とする伝送データ生成装置。

【請求項 8】 前記時間算出手段は、オーバーヘッドが小さな値となるように前記固定ブロックのサイズを設定することを特徴とする請求項 7 に記載する伝送データ生成装置。

【請求項 9】 前記時間算出手段は、セグメントのサイズが固定ブロックのサイズの整数倍でない場合、コンテンツのセグメント毎に、前記固定ブロックで分割できない任意ブロックの再生時間と任意ブロックにおけるオーバーヘッドとを求め、前記セグメント毎のオーバーヘッドを前記固定ブロックにおけるオーバーヘッド及び前記セグメントの任意ブロックにおけるオーバーヘッドに基づいて設定することを特徴とする請求項 7 又は請求項 8 に記載する伝送データ生成装置。

【発明の詳細な説明】

【 0 0 0 1 】

**【発明の属する技術分野】**

本発明は、コンテンツを伝送するための伝送データを生成する伝送データ生成方法及び伝送データ生成装置に関する。

**【0 0 0 2】****【従来の技術】**

近年、インターネットやイントラネットにおいて、A D S L [Asymmetric Digital Subscriber Line]やF T T H [Fiber To The Home]等のブロードバンドの加入者が急速に増加している。ブロードバンドによって、各種コンテンツをストリーミング配信することが可能となり、例えば、V O D [Video On Demand] による個々のユーザに対する映像及び音声の提供やシネマコンプレックスにおいて時間をずらしての同じ映画の提供が可能となった。このようなコンテンツのストリーミング配信では、順方向誤り訂正（F E C [Forward Error Correction]）技術（特許文献 1 参照）やメディア分割技術（特許文献 2 参照）等が利用されている。

**【0 0 0 3】**

順方向誤り訂正技術としては、例えば、ルビー変換（L T [Luby Transform]）がある。ルビー変換では、コンテンツの元データをオーバーヘッドを加味してメタコンテンツ化する。そして、このメタコンテンツ化したデータを持つパッケージが配信される。各パッケージには、コンテンツの元データを求めるための排他的論理和による方程式がランダムに並べられている。受信側では、オーバーヘッド分を含むメタコンテンツ（パッケージ）を受信することができれば、コンテンツの元データをほぼ 1 0 0 % 復元することができ、劣化することなくコンテンツの映像や音声を再生することができる。

**【0 0 0 4】**

メディア分割技術では、ダウンロードを開始してから再生するまでの待ち時間が設定されると、その待ち時間に基づいてコンテンツの先頭からの再生時間とダウンロード時間を算出する。この再生時間とダウンロード時間は、コンテンツの先頭から徐々に増加し、再生時間の累積時間がコンテンツの総再生時間になるまで算出される。そして、ダウンロード時間（又は再生時間）に応じてコンテンツ



がセグメントに分割され、さらに、ルビー変換を組み合わせるとセグメント毎にメタコンテンツ化され、パケット配信される。受信側では、コンテンツの先頭のメタコンテンツ化されたセグメント（パケット）から順次ダウンロードし、ダウンロードしたセグメントを順次再生するすることによってシームレスにコンテンツを再生することができ、非常に短い待ち時間でコンテンツを視聴することができる。

#### 【0005】

##### 【特許文献1】

米国特許出願公開第2001/0019310号明細書

##### 【特許文献2】

米国特許出願公開第2002/0107968号明細書

#### 【0006】

##### 【発明が解決しようとする課題】

しかしながら、メタコンテンツ化する際のオーバーヘッドの最適値は、データのサイズに応じて変動する。ところが、コンテンツやセグメントのサイズは任意のサイズなので、サイズに関係なく、受信側においてコンテンツを劣化することなく再生可能とするために、配信側ではオーバーヘッドを安全側に設定している。つまり、配信側では、どのようなサイズでもコンテンツを100%復元可能とするために、オーバーヘッドの変動範囲（例えば、104～105%）の中で一番大きな値に設定し、冗長データを余分に付加している。そのため、オーバーヘッドをデータサイズに応じて最適値に設定した場合に比べて、メタコンテンツ化した伝送データ量が増加し、伝送帯域が大きくなる。

#### 【0007】

そこで、本発明は、データサイズに応じてオーバーヘッドを最適化できる伝送データ生成方法及び伝送データ生成装置を提供することを課題とする。

#### 【0008】

##### 【課題を解決するための手段】

本発明に係る伝送データ生成方法は、固定ブロックのサイズをオーバーヘッドに基づいて設定する固定ブロックサイズ設定工程と、コンテンツのセグメント毎に

、固定ブロックのサイズに基づいてセグメントのサイズを算出するセグメントサイズ算出工程と、算出したセグメントのサイズに応じてコンテンツをセグメントに分割するセグメント分割工程と、分割したセグメントをブロックに分割するブロック分割工程と、分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ化工程とを含み、固定ブロックにおけるオーバーヘッドに基づいてコンテンツのセグメント毎にオーバーヘッドが設定されることを特徴とする。

#### 【0009】

この伝送データ生成方法では、まず、固定ブロックのサイズをオーバーヘッドに基づいて設定する。続いて、伝送データ生成方法では、コンテンツのセグメント毎に、固定ブロックのサイズに基づいてセグメントのサイズを算出する。そして、伝送データ生成方法では、算出したセグメントのサイズに応じてコンテンツをセグメントに分割し、さらに、分割したセグメントをブロックに分割する。最後に、伝送データ生成方法では、分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化し、ブロック単位の伝送データを生成する。セグメントは、1個以上の固定ブロックからなり、セグメントサイズが大きくなるほど固定ブロックの数が増える。また、セグメントのサイズは、固定ブロックのサイズの整数倍となる場合と整数倍とならない場合がある。したがって、セグメントのオーバーヘッドは、整数倍となる場合には固定ブロックにおけるオーバーヘッドからなり、整数倍とならない場合には固定ブロックにおけるオーバーヘッドと余りのブロックにおけるオーバーヘッドからなる。いずれの場合もセグメントのオーバーヘッドは固定ブロックにおけるオーバーヘッドが基本となっており、整数倍とならない場合でもセグメントのオーバーヘッドはセグメントに含まれる固定ブロックの数が多いほど固定ブロックにおけるオーバーヘッドに近い値となる。したがって、この伝送データ生成方法では、セグメント毎にオーバーヘッドが固定ブロックにおけるオーバーヘッドに基づいて設定される。そこで、この伝送データ生成方法では、固定ブロックのサイズをそのオーバーヘッドを考慮して設定しているので、セグメントのオーバーヘッドをセグメントのサイズ（ひいては、コンテンツのサイズ）に応じて最適な値に設定できる。

## 【0010】

本発明に係る伝送データ生成方法は、固定ブロックのサイズをオーバーヘッドに基づいて設定する固定ブロックサイズ設定工程と、固定ブロックのサイズに基づいて固定ブロックの再生時間を算出する固定ブロック再生時間算出工程と、コンテンツのセグメント毎に、固定ブロックの再生時間に基づいてセグメントの再生時間を算出する再生時間算出工程と、コンテンツのセグメント毎に、算出済みのセグメントの再生時間に基づいてセグメントの伝送時間を算出する伝送時間算出工程と、算出したセグメントの伝送時間に応じてコンテンツをセグメントに分割するセグメント分割工程と、分割したセグメントをブロックに分割するブロック分割工程と、分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ化工程とを含み、固定ブロックにおけるオーバーヘッドに基づいてコンテンツのセグメント毎にオーバーヘッドが設定されることを特徴とする。

## 【0011】

この伝送データ生成方法では、まず、固定ブロックのサイズをオーバーヘッドに基づいて設定し、その設定した固定ブロックのサイズに基づいて固定ブロックの再生時間を算出する。続いて、伝送データ生成方法では、コンテンツのセグメント毎に、算出した固定ブロックの再生時間に基づいてセグメントの再生時間を順次算出し、既に算出しているセグメントの再生時間に基づいてセグメントの伝送時間を算出する。そして、伝送データ生成方法では、算出したセグメントの伝送時間に応じてコンテンツをセグメントに分割し、さらに、分割したセグメントをブロックに分割する。最後に、伝送データ生成方法では、分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化し、ブロック単位の伝送データを生成する。この伝送データ生成方法でも、上記の伝送データ生成方法と同様に、セグメント毎にオーバーヘッドが固定ブロックにおけるオーバーヘッドに基づいて設定され、この固定ブロックのサイズをそのオーバーヘッドを考慮して設定しているので、セグメントのオーバーヘッドをサイズに応じて最適な値を設定できる。ちなみに、実際にセグメントのサイズを設定していく場合には受信側での待ち時間（伝送時間の初期値）に基づいて設定されるので、セグメントや固定ブ

ックのサイズを表すものとして時間（再生時間及び伝送時間）を用いている。

#### 【0012】

なお、セグメントは、コンテンツをメディア分割する際の分割単位であり、待ち時間やチャンネル数等に基づいてセグメント毎の再生時間及び伝送時間が算出され、その伝送時間又は再生時間（セグメントサイズに相当）に応じて分割される。固定ブロックは、コンテンツをメタコンテンツ化及び伝送する際の単位であり、サイズが固定である。固定ブロックのサイズとしては、例えば、64 Mバイト、32 Mバイト、16 Mバイト、8 Mバイト等の64 Mバイトを2の乗数で割ったサイズが設定される。セグメント単位ではなくブロック単位とするのは、サイズの上限が決まったデータを取り扱う必要があるプログラム上の制限やメモリ等のハードウェア上の制限があるからである。ちなみに、セグメント単位でデータを取り扱った場合、データサイズが任意であり、そのサイズの上限も規定できない。

#### 【0013】

ちなみに、セグメントのサイズは、セグメントの伝送時間に伝送速度を乗算した値、また、セグメントの再生時間に再生速度を乗算した値である。固定ブロックのサイズは、固定ブロックの伝送時間に伝送速度を乗算した値、また、固定ブロックの再生時間に再生速度を乗算した値である。したがって、再生時間及び伝送時間は、サイズを表すパラメータであり、時間が長くなるほどサイズが大きくなる。また、再生時間と伝送時間とは比例関係にあり、伝送時間に応じてセグメントに分割することは再生時間に応じてセグメントに分割することと同等である。

#### 【0014】

本発明の上記伝送データ生成方法は、固定ブロックサイズ設定工程では、オーバーヘッドが小さい値となるように固定ブロックのサイズを設定すると好適である。

#### 【0015】

この伝送データ生成方法では、オーバーヘッドが小さい値になるように固定ブロックのサイズを設定することによって、セグメント毎のオーバーヘッドが小さくな

る。したがって、固定ブロック（ひいては、セグメント）毎の伝送データ量が少なくなり、伝送帯域を小さくすることができる。

#### 【0016】

本発明の上記伝送データ生成方法は、セグメントのサイズが固定ブロックのサイズの整数倍でない場合、コンテンツのセグメント毎に、固定ブロックで分割できない任意ブロックのサイズ又は再生時間と任意ブロックにおけるオーバーヘッドとを求める任意ブロック設定工程を含み、セグメントのサイズが固定ブロックのサイズの整数倍でない場合、セグメント毎のオーバーヘッドが、固定ブロックにおけるオーバーヘッド及びセグメントの任意ブロックにおけるオーバーヘッドに基づいて設定される構成とする。

#### 【0017】

この伝送データ生成方法では、セグメントのサイズが固定ブロックのサイズの整数倍とならない場合、コンテンツのセグメント毎に、その固定ブロックで分割できない余りのブロックである任意ブロックのサイズ又は再生時間と任意ブロックのオーバーヘッドを求める。この場合、セグメントのオーバーヘッドは、固定ブロックにおけるオーバーヘッドとセグメント毎の任意ブロックにおけるオーバーヘッドからなる。このように、この伝送データ生成方法では、セグメントが固定ブロックでちょうど分割できない場合でも、その余りのブロックのオーバーヘッドを求めることによって、セグメント毎にオーバーヘッドを設定することができる。

#### 【0018】

なお、任意ブロックは、セグメントを固定ブロックで分割した場合の余りのブロックであり、サイズが任意である。したがって、任意ブロックは、サイズに応じてオーバーヘッドが変わる。ちなみに、任意ブロックのサイズは、任意ブロックの伝送時間に伝送速度を乗算した値、また、任意ブロックの再生時間に再生速度を乗算した値である。

#### 【0019】

本発明の上記伝送データ生成方法は、固定ブロックの再生時間に基づいて固定ブロックの伝送時間を算出する固定ブロック伝送時間算出工程と、コンテンツのセグメント毎に、セグメントの伝送時間及び固定ブロックの伝送時間に基づいて

セグメントに含まれる固定ブロックの数を算出する固定ブロック数算出工程と、コンテンツのセグメント毎に、算出した固定ブロックの数及び固定ブロックの再生時間に基づいてセグメントに含まれる全ての固定ブロックの再生時間を算出する全固定ブロック再生時間算出工程とを含み、再生時間算出工程では、コンテンツのセグメント毎に、セグメントのサイズが固定ブロックのサイズの整数倍の場合にはセグメントに含まれる全ての固定ブロックの再生時間をセグメントの再生時間とし、セグメントのサイズが固定ブロックのサイズの整数倍でない場合にはセグメントの任意ブロックの再生時間及びセグメントに含まれる全ての固定ブロックの再生時間に基づいてセグメントの再生時間を算出する構成とする。

#### 【0020】

この伝送データ生成方法では、固定ブロックの再生時間を算出後、固定ブロックの再生時間に基づいて固定ブロックの伝送時間を算出する。そして、伝送データ生成方法では、コンテンツのセグメント毎に、セグメントの伝送時間と固定ブロックの伝送時間とからセグメントに含まれる固定ブロックの数を算出し、その固定ブロックの数と固定ブロックの再生時間とからセグメントに含まれる全ての固定ブロックの再生時間を算出する。さらに、伝送データ生成方法では、コンテンツのセグメント毎に、セグメントのサイズが固定ブロックのサイズの整数倍の場合にはセグメントに含まれる全ての固定ブロックの再生時間をセグメントの再生時間とし、整数倍とならない場合にはセグメントの任意ブロックの再生時間及びセグメントに含まれる全ての固定ブロックの再生時間に基づいてセグメントの再生時間を算出する。このように、伝送データ生成方法では、セグメントに含まれる固定ブロックの数を簡単に算出することができ、その固定ブロックの数からセグメントの再生時間を算出することができる。

#### 【0021】

本発明の上記伝送データ生成方法は、任意ブロック設定工程では、コンテンツのセグメント毎に、セグメントに含まれる全ての固定ブロックの再生時間及びセグメントの伝送時間を用いて任意ブロックの再生時間と任意ブロックにおけるオーバーヘッドとの乗算値を求め、当該乗算値から所定の数値解析法により任意ブロックの再生時間及び任意ブロックにおけるオーバーヘッドを求める構成とする。

**【0022】**

この伝送データ生成方法では、コンテンツのセグメント毎に、セグメントに含まれる全ての固定ブロックの再生時間を算出すると、その全ての固定ブロックの再生時間及びセグメントの伝送時間を用いて任意ブロックの再生時間とオーバーヘッドとの乗算値を求め、当該乗算値から所定の数値解析法により任意ブロックの再生時間及びオーバーヘッドを求める。この伝送データ生成方法では、任意ブロックの再生時間をそのオーバーヘッドを考慮して求めているので、セグメントのオーバーヘッドをセグメントのサイズ（ひいては、コンテンツのサイズ）に応じて最適な値に設定できる。なお、任意ブロックの再生時間とオーバーヘッドとの2つの変数に対して1つの方程式が成立するが、ニュートン・ラフソン法や二分法等の数値解析法を用いることにより、2つの変数を求めることができる。

**【0023】**

本発明に係る伝送データ生成装置は、固定ブロックのサイズをオーバーヘッドに基づいて設定し、当該固定ブロックのサイズに基づいて固定ブロックの再生時間を算出し、コンテンツのセグメント毎に、当該固定ブロックの再生時間に基づいてセグメントの再生時間を算出し、算出済みのセグメントの再生時間に基づいてセグメントの伝送時間を算出する時間算出手段と、時間算出手段で算出したセグメントの伝送時間に応じてコンテンツをセグメントに分割し、当該分割したセグメントをブロックに分割する分割手段と、分割手段で分割したブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化するメタコンテンツ手段とを備え、時間算出手段は、当該固定ブロックにおけるオーバーヘッドに基づいてセグメント毎にオーバーヘッドを設定することを特徴とする。

**【0024】**

本発明の上記伝送データ生成装置では、時間算出手段は、オーバーヘッドが小さな値となるように固定ブロックのサイズを設定すると好適である。

**【0025】**

本発明の上記伝送データ生成装置では、時間算出手段は、セグメントのサイズが固定ブロックのサイズの整数倍でない場合、コンテンツのセグメント毎に、固定ブロックで分割できない任意ブロックの再生時間と任意ブロックにおけるオー

バヘッドとを求め、セグメント毎のオーバヘッドを固定ブロックにおけるオーバヘッド及びセグメントの任意ブロックにおけるオーバヘッドに基づいて設定する構成とする。

#### 【 0 0 2 6 】

上記伝送データ生成装置では、上記伝送データ生成方法と同様の作用効果を奏する。

#### 【 0 0 2 7 】

##### 【発明の実施の形態】

以下、図面を参照して、本発明に係る伝送データ生成方法及び伝送データ生成装置の実施の形態を説明する。

#### 【 0 0 2 8 】

本実施の形態では、本発明に係る伝送データ生成方法及び伝送データ生成装置を、I P [Internet Protocol]マルチキャスト型V O Dサービスにおけるサーバに適用する。本実施の形態に係るI P [Internet Protocol]マルチキャスト型V O Dサービスでは、サーバがコンテンツをインターネットを介して多数のパーソナルコンピュータ（以下、パソコンと記載する）にストリーミング配信し、パソコンでは短い待ち時間でコンテンツをシームレスに再生することできる。本実施の形態に係るサーバは、V O Dサーバであり、ルビー変換によるメタコンテンツ化とメディア分割を組み合わせることでコンテンツをストリーミング配信する。

#### 【 0 0 2 9 】

図1を参照して、I Pマルチキャスト型V O Dサービスの構成について説明する。図1は、本実施の形態に係るI Pマルチキャスト型V O Dサービスの全体構成図である。なお、図1には、パソコンを1台しか描いていないが、実際には、多数のパソコンが存在する。

#### 【 0 0 3 0 】

I PマルチキャストV O Dサービスでは、サーバ1がインターネットIを介してパソコン2にコンテンツC Nをストリーミング配信する。サーバ1では、コンテンツC Nをメディア分割するとともにメタコンテンツ化してセグメント毎にパケット（伝送データ）を生成することによって、高品質かつ高速配信を実現して



いる。パソコン 2 では、サーバ 1 からのパケットをセグメント毎に順次受信し、受信したパケットをセグメント毎に順次再生することによって、長時間のコンテンツ CN でも短い待ち時間でシームレスに再生することができる。

#### 【 0 0 3 1 】

サーバ 1 では、ハードディスク 1 0 に多数のコンテンツ CN, . . . を格納しており、パケット化するコンテンツ CN をハードディスク 1 0 からドライバ (図示せず) によって読み出す。コンテンツ CN は、少なくとも動画 (映像) のデータを含んでおり、動画のデジタルデータが M P E G [Moving Picture Experts Group] 2 等によって圧縮されて格納されている。コンテンツ CN は、動画の他に音声や文字等のデータを含んでいてもよい。ちなみに、コンテンツ CN の全体のサイズは、コンテンツ CN の最初から最後まで の総再生時間 S と再生するときの速度であるストリーミングレート R とによって表すことができる。

#### 【 0 0 3 2 】

そして、サーバ 1 では、メディア分割部 1 1 によってコンテンツ CN をセグメント S G, . . . に分割し、更に、各セグメント S G をブロック B L, . . . に分割する。さらに、サーバ 1 では、F E C エンコーダ 1 2 によってコンテンツ CN をブロック B L 毎にメタコンテンツ化し、メタコンテンツ化したデータからなるパケット (ブロック単位) を生成する。そして、サーバ 1 では、送信機 1 3 によってパケットをインターネット I を介して配信する。メディア分割部 1 1 及び F E C エンコーダ 1 2 は、専用プログラムをコンピュータで実行することによってソフトウェアで構成される。なお、メディア分割部 1 1 及び F E C エンコーダ 1 2 における処理については、後で詳細に説明する。

#### 【 0 0 3 3 】

パソコン 2 では、受信機 2 0 によってセグメント毎にブロック単位のパケットをダウンロードする。そして、パソコン 2 では、F E C デコーダ 2 1 によってパケットに含まれるメタコンテンツ化されたデータをコンテンツ CN の元データに復元する。さらに、パソコン 2 では、M P E G デコーダ 2 2 によって圧縮化されている元データを解凍する。そして、パソコン 2 では、モニタ 2 3 でセグメント S G 毎にコンテンツ CN を連続再生する。

## 【0034】

メディア分割部 11 及び F E C エンコーダ 12 における処理を説明する前に、メディア分割部 11 及び F E C エンコーダ 12 で用いるパラメータを以下に示す。パソコン 2 でダウンロードを開始してから再生を開始するまでの待ち時間  $w$ 、コンテンツ C N のストリーミングレート  $R$  及び総再生時間  $S$ 、メタコンテンツ化を行う際のオーバーヘッド  $e$ 、サーバ 1 からパソコン 2 にコンテンツ C N を伝送する際のパケットロス耐性  $L$ 、最大受信レート  $R_{\max}$ 、同時受信チャネル数  $C$ 、マルチキャストグループ数  $n$  とする。オーバーヘッド  $e$  とパケットロス耐性  $L$  を加味した実効ストリーミングレート  $R_{\text{eff}}$  は、式 (1) により求めることができ、冗長度を加味したコンテンツのストリーミングレートを示す。また、ストリーミングレート比  $m$  は、式 (2) により求めることができ、最大受信レート  $R_{\max}$  に対する実効ストリーミングレート  $R_{\text{eff}}$  を示す。また、チャネル当たりの送信レート  $R_s$  は、式 (3) により求めることができる。

## 【0035】

## 【数 1】

$$R_{\text{eff}} = \frac{e \times R}{1 - L} \dots (1)$$

$$m = \frac{R_{\max}}{R_{\text{eff}}} = \frac{R_{\max} \times (1 - L)}{e \times R} \dots (2)$$

$$R_s = \frac{R_{\max}}{C} = \frac{m \times R_{\text{eff}}}{C} \dots (3)$$

なお、パケットロス耐性  $L$  は、単位が百分率 [%] で表されるが、本実施の形態で用いられる各式では百分率に応じた 1 以下の数値（例えば、10% の場合には 0.1）が用いられる。また、オーバーヘッド  $e$  も、単位が百分率 [%] で表されるが、本実施の形態で用いられる各式では百分率に応じた 1.04 ~ 1.05 の数値が用いられる。

## 【0036】

図 1 ~ 図 10 を参照して、メディア分割部 11 における処理を詳細に説明する。図 2 は、メディア分割部における処理の説明図である。図 3 は、同時受信チャ

ネル数が1の場合のダウンロード時間と再生時間の関係を示す図である。図4は、同時受信チャネル数が2の場合のダウンロード時間と再生時間の関係を示す図である。図5は、同時受信チャネル数が3の場合のダウンロード時間と再生時間の関係を示す図である。図6は、ダウンロード時間及び再生時間とストリーミングレートとの関係を示す図である。図7は、メディア分割部におけるブロック分割の説明図である。図8は、ルビー変換におけるデータサイズとオーバーヘッドとの関係を示す図である。図9は、オーバーヘッドを最適化した場合のダウンロード時間及び再生時間とストリーミングレートとの関係を示す図である。図10は、ニュートン・ラフソン法の説明図ある。

#### 【0037】

メディア分割部11では、図2に示すように、コンテンツCNをセグメントSGに分割し、さらに、そのセグメントSGを固定ブロックBLfと固定ブロックBLfで分割できなかった余りのブロックである任意ブロックBLoに分割する。セグメントSGは、設定された待ち時間wでシームレスにストリーミング配信するために、コンテンツCNをメディア分割する際の分割単位である。セグメントSGのサイズは、任意のサイズであり、コンテンツCNの先頭から徐々に増加していく。固定ブロックBLfは、コンテンツCNをメタコンテンツ化及び伝送する際の単位であり、サイズが固定である。固定ブロックBLfのサイズは、再生装置（パソコン2）のメモリに読み込む必要があるため、その制約によって決まる。任意ブロックBLoは、セグメントSGを固定ブロックBLfで分割した場合に固定ブロックBLfで分割できなかった余りのブロックである。任意ブロックBLoのサイズは、固定ブロックBLfのサイズより小さい任意のサイズである。ちなみに、セグメントSGを固定ブロックBLfでちょうど分割できた場合、そのセグメントSGには任意ブロックBLoは存在しない。

#### 【0038】

セグメントSGに分割するのは、受信側においてセグメント単位でダウンロード及び再生を繰り返し実行することによって、ダウンロードを開始してから設定された待ち時間wを待つだけでシームレスな再生を可能とするためである。セグメントSGのサイズは、ダウンロード時間 $b_1$ ,  $b_2$ , ...と送信レート（伝送

速度)との乗算値又は再生時間  $a_1, a_2, \dots$  とストリーミングレート(再生速度)との乗算値で表すことができる。送信レート及びストリーミングレートはそれぞれ一定の速度なので、セグメント  $SG, \dots$  はそのサイズを相当するダウンロード時間  $b_1, b_2, \dots$  に応じて分割される。再生時間  $a_1, a_2, \dots$  を全て加算した時間は、コンテンツ  $CN$  の総再生時間  $S$  になる。ちなみに、ダウンロード時間  $b_1, b_2, \dots$  と再生時間  $a_1, a_2, \dots$  とは比例関係にあり、ダウンロード時間  $b_1, b_2, \dots$  に応じた分割は再生時間  $a_1, a_2, \dots$  に応じた分割と同じ分割結果が得られる。

#### 【0039】

セグメント  $SG$  をブロック  $BL_f, BL_o$  に分割するのは、サーバ1及びパソコン2で取り扱うデータ量の上限を規定するためである。というのは、サーバ1及びパソコン2には、取り扱うことができるデータ量に、プログラム上の制限やメモリ等のハードウェア上の制限があるからである。

#### 【0040】

図3～図5を参照して、メディア分割部11における再生時間とダウンロード時間によるセグメント分割の方法を、3つの例を挙げて説明する。図3～図5に示す例では、待ち時間  $w$  を30秒、ストリーミングレート  $R$  を600 kbps、オーバーヘッド  $e$  を1.05(105%)、パケットロス耐性  $L$  を10%、最大受信レート  $R_{\max}$  を1400 kbpsとしている。この場合、実効ストリーミングレート  $R_{\text{eff}}$  は、式(1)により、700 kbpsである。また、ストリーミングレート比  $m$  は、式(2)により、2である。

#### 【0041】

まず、メディア分割部11では、ダウンロード時間の初期値として待ち時間  $w = 30$  秒(=0.5分)を1つ目のセグメントのダウンロード時間  $b_1$  に設定する。そして、メディア分割部11では、各セグメントの再生時間  $a_n$  を順次演算し、算出済みの1つ又は複数の再生時間  $a_n, a_{n-1}, \dots$  から次のセグメントのダウンロード時間  $b_{n+1}$  を順次演算する。この際、連続するセグメント間で再生が途切れないように再生時間を設定するとともに、1つ又は複数のセグメントの再生時間中に次のセグメントのダウンロード時間を設定する。また、再生時間

$a_1, \dots, a_n$ の積算時間が総再生時間  $S$  になるまで、再生時間とダウンロード時間とを順次設定する。ちなみに、待ち時間  $w$  は、パソコン 2 からユーザが設定してもよいし、あるいは、サーバ 1 側で予め設定しておいてもよい。

#### 【0042】

図 3 の例では、同時受信チャネル数  $C$  が 1 である。そこで、1 つのセグメントの再生時間に対して次のセグメントのダウンロード時間を設定する。また、チャネル当たりの送信レート  $R_s$  は、式 (3) により  $1400 \text{ kbps}$  であり、実効ストリーミングレート  $R_{\text{eff}}$  の 2 倍である。したがって、各セグメントにおいてダウンロード時間に対して再生時間が 2 倍となる。まず、1 つ目のセグメントのダウンロード時間を 0.5 分間に設定すると、0.5 分後に、2 つ目のセグメントのダウンロード時間を 1 分間設定し、1.5 分後に、3 つ目のセグメントのダウンロード時間を 2 分間設定する。

#### 【0043】

図 4 の例では、同時受信チャネル数  $C$  が 2 である。そこで、2 つのセグメントの再生時間に対して次のセグメントのダウンロード時間を設定する。また、チャネル当たりの送信レート  $R_s$  は、式 (3) により  $700 \text{ kbps}$  であり、実効ストリーミングレート  $R_{\text{eff}}$  の 1 倍である。したがって、各セグメントにおいてダウンロード時間に対して再生時間が 1 倍となる。まず、1 つ目のセグメントのダウンロード時間を 0.5 分間及び再生時間を 0.5 分間設定するとともに 2 つ目のセグメントのダウンロード時間を 1 分間設定すると、0.5 分後に、3 つ目のセグメントのダウンロード時間を 1.5 ( $= 0.5 + 1$ ) 分間設定し、1 分後に、4 つ目のセグメントのダウンロード時間を 2.5 ( $= 1 + 0.5$ ) 分間設定し、2 分後に、5 つ目のセグメントのダウンロード時間を 4.0 ( $= 1.5 + 2.5$ ) 分間設定する。

#### 【0044】

図 5 の例では、同時受信チャネル数  $C$  が 3 である。そこで、3 つのセグメントの再生時間に対して次のセグメントのダウンロード時間を設定する。また、チャネル当たりの実効送信レート  $R_s$  は、式 (3) により  $467 \text{ kbps}$  であり、実効ストリーミングレート  $R_{\text{eff}}$  の  $2/3$  倍である。したがって、各セグメントに

においてダウンロード時間に対して再生時間が2/3倍となる。まず、1つ目のセグメントのダウンロード時間を0.5分間及び再生時間を0.33分間設定するとともに2つ目のセグメントのダウンロード時間を0.83分間及び再生時間を0.56分間設定するとともに3つのセグメントのダウンロード時間を1.39分間設定すると、0.5分後に、4つ目のセグメントのダウンロード時間を1.82(=0.33+0.56+0.93)分間設定し、0.83分後に、5つ目のセグメントのダウンロード時間を2.70(=0.56+0.93+1.21)分間設定し、1.39分後に、6つ目のセグメントのダウンロード時間を3.93(=0.93+1.21+1.79)分間設定する。

#### 【0045】

ここで、 $n$ 番目のセグメントのダウンロード時間を $b_n$ とし、再生時間を $a_n$ とした場合、ダウンロード時間 $b_n$ と再生時間 $a_n$ との関係は式(4)に示す関係となる。メディア分割部11では、ダウンロード時間 $b_n$ を、式(5)又は式(6)により、 $n$ 番目以前の再生時間 $a_{n-1}$ ,  $a_{n-2}$ , ... から演算する。

#### 【0046】

##### 【数2】

$$a_n = \frac{m}{C} \times b_n \cdots (4)$$

$$n - C < 1 \text{ の場合 } b_n = b_1 + \sum_{i=1}^{n-1} a_i \cdots (5)$$

$$n - C \geq 1 \text{ の場合 } b_n = \sum_{i=n-C}^{n-1} a_i \cdots (6)$$

式(5)、(6)に示すように、セグメントのダウンロード時間 $b_n$ は既に演算されている再生時間 $a_{n-1}$ ,  $a_{n-2}$ , ... を加算して求めることができ、チャネル数 $C$ によってその加算する個数が変わる。

#### 【0047】

図6を参照して、各セグメントにおける再生時間 $a_n$ 及びダウンロード時間 $b_n$ とストリーミングレート $R$ 及び実効ストリーミングレート $R_{eff}$ との関係について説明する。各セグメントでは、再生データとダウンロードデータとは同じサイ

ズなので、式(7)に示すように、再生時間  $a_n$  とストリーミングレート  $R$  との乗算値(図6の白塗りつぶし領域参照)はダウンロード時間  $b_n$  とチャンネル当たりの送信レート  $mR/C$  との乗算値に等しくなる(図6の斜線領域参照)。さらに、パソコン2側でコンテンツ  $CN$  の元データに100%復元可能とするために、サーバ1側では、各セグメントにおいてメタコンテンツ化する際のオーバーヘッド  $e$  (図6の  $e$  で示す点領域参照)と伝送中のパケットロスに対するパケットロス耐性  $L$  (図6の  $1/(1-L)$  で示す点領域参照)による冗長データ分を加味している。したがって、各セグメントでは、式(8)に示すように、再生時間  $a_n$  と実効ストリーミングレート  $R_{eff}$  との乗算値はダウンロード時間  $b_n$  とチャンネル当たりの実効送信レート  $mR_{eff}/C$  との乗算値に等しくなる。なお、オーバーヘッド  $e$  は、出来るだけ小さな最適な値が設定され、例えば、104%~105%の値である。パケットロス耐性  $L$  は、サーバ1とパソコン2との間のネットワークにおいて想定される値が設定され、下限が0%であり、上限が50%未満であり、例えば、10%程度である。

【0048】

【数3】

$$R \times a_n = \frac{m \times R}{C} \times b_n \cdots (7)$$

$$R_{eff} \times a_n = \frac{m \times R_{eff}}{C} \times b_n \cdots (8)$$

式(7)、(8)では、時間とレートとの乗算値であるデータサイズが、再生データとダウンロードデータとで変わらないことを示している。

【0049】

一方、パソコン2側では、伝送中にパケットロス耐性  $L$  に相当するパケットが損失しているので、ダウンロードデータからパケットロス耐性  $L$  に相当する冗長データが減少している。さらに、パソコン2側では、メタコンテンツ化されたデータを元データに復元する際にオーバーヘッド  $e$  に相当する冗長データを損失するので、再生データにおいてオーバーヘッド  $e$  に相当する冗長データが減少している。

。

## 【0050】

図2、図7～図10を参照して、メディア分割部11におけるブロック分割の方法を説明する。まず、メディア分割部11では、固定ブロック $B_L f$ のサイズを設定する（図2参照）。固定ブロック $B_L f$ のサイズ $B_s$ は、固定ブロックのオーバーヘッド $e_1$ の値が小さな値になるように設定する。メタコンテンツ化におけるオーバーヘッドは、図8に示すように、104～105%の間をデータサイズに応じて変動している。シミュレーションや実験等の結果、64Mバイトを2の乗数で割ったデータサイズ（64Mバイト、32Mバイト、16Mバイト、・・・）においてオーバーヘッド $e$ が周期的に小さくなる。そこで、メディア分割部11では、固定ブロック $B_L f$ のサイズ $B_s$ として、例えば、オーバーヘッド $e_1$ が104%となる64Mバイトを設定する。この場合、サーバ1において64Mバイトの元データを10%のオーバーヘッドを加味してメタコンテンツ化し、パソコン2においてこの4%分の冗長データを含むメタコンテンツデータを受信できれば、元データをほぼ100%復元できる。このように、サーバ1では、固定ブロックのオーバーヘッド $e_1$ を極力小さな値に設定し、冗長データ量を少なくし、固定ブロックにおけるデータ量を低減している。

## 【0051】

次に、メディア分割部11では、式（9）により、固定ブロックサイズ $B_s$ から固定ブロックの再生時間 $d_a$ を演算する（図7参照）。続いて、メディア分割部11では、式（10）により、固定ブロックのオーバーヘッド $e_1$ から固定ブロックにおけるストリーミングレート比 $m_1$ を演算する。さらに、メディア分割部11では、式（11）により、固定ブロックの再生時間 $d_a$ とストリーミングレート比 $m_1$ から固定ブロックのダウンロード時間 $d_b$ を演算する（図7参照）。なお、ダウンロード時間 $d_b$ がセグメントのダウンロード時間 $b_n$ より大きくなった場合には、固定ブロックサイズ $B_s$ を半分のサイズに再設定する。さらに、半分にしても大きい場合には、更に半分にする。

## 【0052】



## 【数 4】

$$d_a = \frac{B_s}{R} \cdot \cdot \cdot (9)$$

$$m_1 = \frac{R_{max} \times (1 - L)}{e_1 \times R} \cdot \cdot \cdot (10)$$

$$d_b = \frac{C}{m_1} \times d_a \cdot \cdot \cdot (11)$$

この固定ブロックの再生時間  $d_a$  及びダウンロード時間  $d_b$  は、固定ブロックサイズ  $B_s$  が全てのセグメントに対して共通なので、全てのセグメントにおいて同じ値である。

## 【0053】

そして、メディア分割部 11 では、セグメント毎に、式 (12) により、各セグメントに含まれる固定ブロックの数  $k$  を演算する (図 7 参照)。続いて、メディア分割部 11 では、セグメント毎に、式 (13) により、各セグメントに含まれる全ての固定ブロックによる全再生時間  $a_{n1}$  を演算する (図 7 参照)。さらに、メディア分割部 11 では、セグメント毎に、式 (14) により各セグメントに含まれる全ての固定ブロックによる全ダウンロード時間  $b_{n1}$  を演算し、式 (15) により各セグメントの任意ブロックにおけるダウンロード時間  $b_{n2}$  を演算する (図 7 参照)。

## 【0054】

## 【数 5】

$$k = \text{INT} \left[ \frac{b_n}{d_b} \right] \cdot \cdot \cdot (12)$$

$$a_{n1} = k \times d_a \cdot \cdot \cdot (13)$$

$$b_{n1} = k \times d_b \cdot \cdot \cdot (14)$$

$$b_{n2} = b_n - b_{n1} = b_n - k \times d_b \cdot \cdot \cdot (15)$$

なお、 $\text{INT} []$  は整数化関数であり、分子を分母で除算したときの除算値の小数点以下を切り捨てた整数値を求めることができる。

## 【0055】

ここで、セグメントにおける任意ブロックの再生時間  $a_{n2}$  が判らないと、セグメントの再生時間  $a_n$  を求めることができない。また、セグメントにおける任意ブロックのオーバーヘッド  $e_2$  が判らないと、セグメントにおけるオーバーヘッド  $e$  が決まらない。そこで、メディア分割部 11 では、セグメント毎に、任意ブロックの再生時間  $a_{n2}$  及びオーバーヘッド  $e_2$  を求める。

#### 【0056】

固定ブロックのオーバーヘッド  $e_1$  はオーバーヘッドの変動範囲の中でも小さい値に固定されているが、任意ブロックのオーバーヘッド  $e_2$  は、任意ブロックのサイズに応じて変動するので、オーバーヘッド  $e_1$  より大きくなっている（図9参照）。ダウンロードデータと再生データとのサイズは同じなので、セグメントに含まれる全ての固定ブロックの全再生時間  $a_{n1}$  と全ダウンロード時間  $b_{n1}$  とは式（16）に示す関係となり、セグメントの任意ブロックの再生時間  $a_{n2}$  とダウンロード時間  $b_{n2}$  とは式（17）に示す関係となる（図9参照）。式（16）と式（17）から、セグメントのダウンロード時間  $b_n$  と固定ブロックの全再生時間  $a_{n1}$  及び任意ブロックの再生時間  $a_{n2}$  とは式（18）に示す関係となる。また、オーバーヘッド  $e_1$  とパケットロス耐性  $L$  を加味した固定ブロックにおける実効ストリーミングレート  $R_{eff1}$  は式（19）により演算することができ、オーバーヘッド  $e_2$  とパケットロス耐性  $L$  を加味した任意ブロックにおける実効ストリーミングレート  $R_{eff2}$  は式（20）により演算することができる（図9参照）。

#### 【0057】

##### 【数6】

$$b_{n1} = \frac{C}{m_1} \times a_{n1} \cdots (16)$$

$$b_{n2} = \frac{C}{m_2} \times a_{n2} \cdots (17)$$

$$b_n = b_{n1} + b_{n2} = \frac{C}{m_1} \times a_{n1} + \frac{C}{m_2} \times a_{n2} \cdots (18)$$

$$R_{eff1} = \frac{e_1 \times R}{1 - L} \cdots (19)$$

$$R_{eff2} = \frac{e_2 \times R}{1 - L} \cdots (20)$$

したがって、固定ブロックにおけるストリーミングレート比 $m_1$ は式(21)によって演算でき、任意ブロックにおけるストリーミングレート比 $m_2$ は式(22)によって演算できる。このストリーミングレート比 $m_1$ ,  $m_2$ を式(18)に代入すると、式(23)となる。この式(23)を式(24)に変形し、任意ブロックにおけるオーバーヘッド $e_2$ と再生時間 $a_{n2}$ との乗算値を演算する式とする。

【0058】

【数7】

$$m_1 = \frac{R_{max}}{R_{eff1}} = \frac{R_{max} \times (1-L)}{e_1 \times R} \dots (21)$$

$$m_2 = \frac{R_{max}}{R_{eff2}} = \frac{R_{max} \times (1-L)}{e_2 \times R} \dots (22)$$

$$b_n = \frac{e_1 \times R \times C}{R_{max} \times (1-L)} \times a_{n1} + \frac{e_2 \times R \times C}{R_{max} \times (1-L)} \times a_{n2}$$

$$= \frac{R \times C}{R_{max} \times (1-L)} (e_1 \times a_{n1} + e_2 \times a_{n2}) \dots (23)$$

$$e_2 a_{n2} = \frac{b_n \times R_{max} \times (1-L)}{R \times C} - e_1 \times a_{n1} \dots (24)$$

式(13)により固定ブロックによる全再生時間 $a_{n1}$ を演算でき、式(5)、(6)によりセグメントのダウンロード時間 $b_n$ を演算でき、固定ブロックのオーバーヘッド $e_1$ は固定ブロックサイズ $B_s$ を設定する際に確定している。したがって、式(24)の右辺の全てのパラメータは既知であり、その右辺を演算することができる。つまり、任意ブロックにおけるオーバーヘッド $e_2$ と再生時間 $a_{n2}$ との乗算値は、セグメント毎に演算でき、ある一定値となる。

【0059】

したがって、任意ブロックにおけるオーバーヘッド $e_2$ は再生時間 $a_{n2}$ の関数となり、式(25)に示す関係となる。式(24)と式(25)から、式(26)が成立する。

【0060】

## 【数 8】

$$e_2 = f(a_{n2}) \cdots (25)$$

$$f(a_{n2}) = \frac{\text{一定値}}{a_{n2}} \cdots (26)$$

ここで、任意ブロックにおける再生時間  $a_{n2}$  とオーバヘッド  $e_2$  とを、ニュートン・ラフソン法を利用することによって一意に確定することができる（図 10 参照）。そこで、メディア分割部 11 では、式（24）の右辺を演算後、ニュートン・ラフソン法を用いて任意ブロックにおける再生時間  $a_{n2}$  とオーバヘッド  $e_2$  を求める。ニュートン・ラフソン法では、オーバヘッド  $e_2 (= f(a_{n2}))$  の初期値として一意に決まる値に近い値（例えば、104～105%の中央値の104.5%）に設定し、その初期値から徐々に収束させていき、オーバヘッド  $e_2$  を求める。このように、サーバ 1 では、任意ブロックのオーバヘッド  $e_2$  をセグメント毎に最適な値を設定し、冗長データ量を少なくし、任意ブロックにおけるデータ量を低減している。

## 【0061】

## 【数 9】

$$a_n = a_{n1} + a_{n2} \cdots (27)$$

そして、メディア分割部 11 では、セグメント毎に、式（27）により、固定ブロックの全再生時間  $a_{n1}$  と任意ブロックの再生時間  $a_{n2}$  とからセグメントの再生時間  $a_n$  を演算する。

## 【0062】

## 【数 10】

$$e_{ave} = \frac{e_1 \times a_{n1} + e_2 \times a_{n2}}{a_{n1} + a_{n2}} \cdots (28)$$

$$m_{ave} = \frac{R_{max} \times (1-L)}{e_{ave} \times R} \cdots (29)$$

$$a_n = \frac{m_{ave}}{C} \times b_n = \frac{R_{max} \times (1-L)}{e_{ave} \times R \times C} \times b_n \cdots (30)$$

あるいは、メディア分割部 11 では、式（28）によりセグメントにおけるオ

ーバヘッドの平均値  $e_{ave}$  や式 (29) によりセグメントにおけるストリーミングレート比の平均値  $m_{ave}$  を演算する。そして、メディア分割部 11 では、式 (30) により、そのオーバヘッドの平均値  $e_{ave}$  やストリーミングレート比の平均値  $m_{ave}$  を用いてセグメントの再生時間  $a_n$  を演算する。

#### 【0063】

このように、メディア分割部 11 では、各セグメントの再生時間  $a_n$  を演算し、既に演算したセグメントの再生時間  $a_n$ ,  $a_{n-1}$ , ... から次のセグメントのダウンロード時間  $b_{n+1}$  を演算する。その演算過程で、まず、固定ブロックのサイズ  $B_s$  及びオーバヘッド  $e_1$  を設定し、固定ブロックの再生時間  $d_a$  及びダウンロード時間  $d_b$  を演算する。次に、セグメント毎に、セグメントに含まれる全ての固定ブロックの全再生時間  $a_{n1}$  及び全ダウンロード時間  $b_{n1}$ 、任意ブロックのダウンロード時間  $b_{n2}$  を演算する。さらに、セグメント毎に、任意ブロックのオーバヘッド  $e_2$  及び再生時間  $a_{n2}$  を求める。

#### 【0064】

そして、メディア分割部 11 では、各セグメントのダウンロード時間  $b_n$  (又は再生時間  $a_n$ ) に応じてコンテンツ  $CN$  をセグメント  $SG$ , ... に分割する (図 2 参照)。さらに、メディア分割部 11 では、セグメント毎に、固定ブロックのダウンロード時間  $d_b$  (又は再生時間  $d_a$ ) 及び任意ブロックのダウンロード時間  $b_{n2}$  (又は再生時間  $a_{n2}$ ) で固定ブロック  $BLf$ , ... 及び任意ブロック  $BLo$  に分割する (図 2 参照)。この分割は、各セグメントのダウンロード時間  $b_n$  を演算する毎に行ってもよいし、あるいは、全てセグメントのダウンロード時間  $b_1$ ,  $b_2$ , ... を演算した後に行ってもよい。

#### 【0065】

なお、各セグメントにおけるオーバヘッド  $e$  は、式 (28) で演算されるオーバヘッドの平均値  $e_{ave}$  であり、セグメントのサイズに応じた最適な値となっている。オーバヘッドの平均値  $e_{ave}$  は、固定ブロックのオーバヘッド  $e_1$  (= 104%) の影響を受け、104.5% 以下となり、各セグメントに含まれる固定ブロック数  $k$  が多いほど 104% に近い値となる。

#### 【0066】

図 1、図 1 1 及び図 1 2 を参照して、F E C エンコーダ 1 2 における処理を詳細に説明する。図 1 1 は、F E C エンコーダにおける処理の説明図である。図 1 2 は、メタコンテンツ化の説明図である。

#### 【0067】

F E C エンコーダ 1 2 では、図 1 1 に示すように、セグメント S G、・・・毎に、コンテンツ C N の元データをブロック B L 単位でメタコンテンツ化する。この際、メタコンテンツ化したデータには、オーバーヘッド e 及びパケットロス耐性 L に対する冗長データ（図 1 1 の点領域）が加味されている。さらに、F E C エンコーダ 1 2 では、図 1 1 に示すように、ブロック B L 単位のメタコンテンツデータに I P パケットヘッダを付加し（図 1 1 の横線領域）、ブロック単位のパケットを生成する。パケット化後のデータサイズは、メタコンテンツ化後のデータサイズの例えば（1086 / 1024）倍となる。

#### 【0068】

図 1 2 を参照して、ルビー変換によるメタコンテンツ化について説明する。図 1 2 では、コンテンツの元データに相当する入力データが a ~ h の 10 個ある例を示している。ルビー変換では、各入力データ a ~ h に対して乱数を発生させる。そして、出力データとして各列に対して排他的論理和による方程式（例えば、一列目の場合には  $a \text{ XOR } g$ ）をたて、この方程式がメタコンテンツデータとなる。

#### 【0069】

基本的には、入力データの個数分の方程式（出力データ）が有れば、方程式を解くことができ、入力データを復元できる。しかし、ルビー変換では、このメタコンテンツデータをランダムに並び替え、このランダムに並び替えられたメタコンテンツデータの中に冗長データが含まれるようにしている。したがって、入力データの個数分の方程式（出力データ）では、入力データを 100% 復元することができない。そこで、ルビー変換では、オーバーヘッド e を設定し、入力データの個数にオーバーヘッド e を加味した個数の方程式（出力データ）からメタコンテンツを生成する。例えば、オーバーヘッド e を 104% とすると、100 個の入力データに対して 104 個の出力データ（方程式）からなるメタコンテンツとなる。

。

#### 【0070】

図1を参照して、サーバ1における動作を説明する。特に、メディア分割部11における再生時間及びダウンロード時間の演算処理については図13のフローチャートに沿って説明する。図13は、本実施の形態に係る再生時間及びダウンロード時間の演算処理を示すフローチャートである。

#### 【0071】

サーバ1では、あるコンテンツCNをストリーミング配信するために、コンテンツCNをハードディスク10から読み出す。各コンテンツCNはコンテンツサイズCs、ストリーミングレートR、総再生時間S等のコンテンツ情報を有しており、これらの情報もハードディスク10に格納されている。

#### 【0072】

サーバ1では、メディア分割するために、まず、コンテンツ情報を読み込む(S1)。次に、サーバ1では、待ち時間w、パケットロス耐性L、最大受信レート $R_{max}$ 、チャネル数C等の配信情報を取得する(S2)。配信情報の取得は、予め情報を取得してハードディスク10に格納し、ハードディスク10から読み込んでもよいし、あるいは、パソコン2とインターネットIを介して接続した際に情報を取得してもよい。

#### 【0073】

各種情報を取得すると、サーバ1では、固定ブロックのオーバーヘッド $e_1$ が出来るだけ小さい値となるように、固定ブロックサイズ $B_s$ とオーバーヘッド $e_1$ を設定する(S3)(図8参照)。そして、サーバ1では、式(9)により、固定ブロックサイズ $B_s$ から固定ブロックの再生時間 $d_a$ を演算する(S4)(図7参照)。さらに、サーバ1では、式(10)により、オーバーヘッド $e_1$ から固定ブロックのストリーミングレート比 $m_1$ を演算する(S5)。続いて、サーバ1では、式(11)により、再生時間 $d_a$ 及びストリーミングレート比 $m_1$ から固定ブロックのダウンロード時間 $d_b$ を演算する(S6)(図7参照)。

#### 【0074】

次に、サーバ1では、コンテンツCNの先頭から何番目のセグメントに対する

処理かを示す  $n$  に 1 を設定する (S 7)。そして、サーバ 1 では、1 番目のセグメントのダウンロード時間  $b_1$  (初期値) として待ち時間  $w$  を設定する (S 8)。ちなみに、ここまでの処理は初期設定として一度だけ実行され、以下の処理はセグメント毎に繰り返し実行される。

#### 【0075】

サーバ 1 では、式 (12) により、セグメントのダウンロード時間  $b_n$  及び固定ブロックのダウンロード時間  $d_b$  から固定ブロック数  $k$  を演算する (S 9) (図 7 参照)。ダウンロード時間  $b_n$  は、 $n$  が 1 の場合には S 8 の処理で初期設定され、 $n$  が 1 以外の場合には S 18 又は S 19 の処理で演算されている。続いて、サーバ 1 では、式 (13) により、固定ブロックの再生時間  $d_a$  及び固定ブロック数  $k$  からセグメントに含まれる全ての固定ブロックの全再生時間  $a_{n1}$  を演算する (S 10) (図 7 参照)。

#### 【0076】

サーバ 1 では、ダウンロード時間  $b_n$ 、固定ブロックのオーバーヘッド  $e_1$  及び全再生時間  $a_{n1}$  等を用いて、式 (24) の右辺値 (すなわち、任意ブロックのオーバーヘッド  $e_2$  と再生時間  $a_{n2}$  との乗算値) を演算する。そして、サーバ 1 では、ニュートン・ラフソン法により、この式 (24) の右辺値を用いて任意ブロックのオーバーヘッド  $e_2$  及び再生時間  $a_{n2}$  を求める (S 11) (図 10 参照)。

#### 【0077】

サーバ 1 では、式 (28) により、固定ブロックのオーバーヘッド  $e_1$  及び任意ブロックのオーバーヘッド  $e_2$  からセグメントにおけるオーバーヘッドの平均値  $e_{ave}$  を演算する (S 12)。また、サーバ 1 では、式 (29) により、オーバーヘッドの平均値  $e_{ave}$  からセグメントにおけるストリーミングレート比の平均値  $m_{ave}$  を演算する (S 13)。

#### 【0078】

そして、サーバ 1 では、式 (30) により、オーバーヘッドの平均値  $e_{ave}$  又はストリーミングレート比の平均値  $m_{ave}$  からセグメントの再生時間  $a_n$  を演算する (S 14)。あるいは、サーバ 1 では、式 (27) により、固定ブロックの全再生時間  $a_{n1}$  及び任意ブロックの再生時間  $a_{n2}$  からセグメントの再生時間  $a_n$  を演



算する (S14)。

【0079】

ダウンロード時間及び再生時間の演算処理を終了するか否かを判断するために、サーバ1では、これまで演算した再生時間  $a_1 \sim a_n$  の積算時間がコンテンツ C N の総再生時間 S 未満か否かを判定する (S15)。積算時間が総再生時間 S 以上と判定した場合、サーバ1では、ダウンロード時間及び再生時間の演算処理を終了する。

【0080】

一方、積算時間が総再生時間 S 未満と判定した場合、サーバ1では、 $n$  に 1 を加算して次のセグメントに対する演算処理に移る (S16)。

【0081】

そして、サーバ1では、( $n$  - チャネル数 C) が 1 未満か否かを判定する (S17)。(  $n$  - チャネル数 C) が 1 未満と判定した場合には、サーバ1では、式 (5) により、ダウンロード時間の初期値  $b_1$  及び既に演算した再生時間  $a_1, a_2, \dots$  から次のセグメントのダウンロード時間  $b_n$  を演算する (S18)。一方、( $n$  - チャネル数 C) が 1 以上と判定した場合には、サーバ1では、式 (6) により、既に演算した再生時間  $a_{n-1}, a_{n-2}, \dots$  から次のセグメントのダウンロード時間  $b_n$  を演算する (S19)。ダウンロード時間  $b_n$  を演算すると、サーバ1では、S9 に戻って処理を続ける。

【0082】

コンテンツ C N に対するダウンロード時間及び再生時間の演算処理が終了すると、サーバ1では、各セグメントのダウンロード時間  $b_n$  (又は再生時間  $a_n$ ) に応じてコンテンツ C N をセグメント S G に順次分割する (図2参照)。さらに、サーバ1では、セグメント S G 毎を、固定ブロックのダウンロード時間  $d_b$  (又は再生時間  $d_a$ ) 及び任意ブロックのダウンロード時間  $b_{n2}$  (又は再生時間  $a_{n2}$ ) に応じて固定ブロック B L f,  $\dots$  及び任意ブロック B L o に分割する (図2参照)。

【0083】

続いて、サーバ1では、セグメント S G 毎に、コンテンツ C N の元データをブ

ロック B L 単位でメタコンテンツ化する（図 1 1、図 1 2 参照）。さらに、サーバ 1 では、ブロック B L 単位のメタコンテンツデータをパケット化する（図 1 1 参照）。

#### 【0084】

そして、サーバ 1 では、セグメント S G 毎に、ブロック単位のパケットをインターネット I を介してパソコン 2 にストリーミング配信する（図 1 参照）。

#### 【0085】

なお、コンテンツ C N に対するメディア分割や F E C エンコードは、ストリーミング配信する度に行う必要はなく、一度行えばよい。

#### 【0086】

図 1 4 には、オーバーヘッド  $e$  の違いにより、マルチキャスト帯域がどの程度変わるのかをパケットロス耐性に応じて示している。ここでの条件は、ストリーミングレート  $R$  が  $6000\text{ kbps}$ 、総再生時間  $S$  が  $100$  分、最大受信レート  $R_{\max}$  が  $12000\text{ kbps}$ 、待ち時間  $w$  が  $20$  秒、同時受信チャネル数が  $3$  又は  $8$  とする。また、オーバーヘッド  $e$  は、 $105\%$  と  $104\%$  とする。図 1 4 から判るように、オーバーヘッド  $e$  が  $105\%$  の場合のマルチキャスト帯域（一点鎖線と二点鎖線）より、オーバーヘッド  $e$  が  $104\%$  の場合のマルチキャスト帯域（太い実線と細い実線）が小さくなっている。この傾向は、パケットロス耐性  $L$  が大きくなるほど顕著になっている。サーバ 1 では、固定ブロックのオーバーヘッド  $e_1$  を  $104\%$  とし、セグメント毎のオーバーヘッド  $e$  が  $104\%$  に極力近づくように最適化している。そのため、サーバ 1 によるセグメント毎に最適化したオーバーヘッド  $e$  の場合のマルチキャスト帯域は、従来の安全側に一律に設定されていたオーバーヘッド  $e$ （例えば、 $105\%$ ）の場合のマルチキャスト帯域より小さくなる。

#### 【0087】

このサーバ 1 によれば、固定ブロックのオーバーヘッド  $e_1$  が極力小さくなるように固定ブロックサイズ  $B_s$  を設定するとともに任意ブロックのオーバーヘッド  $e_2$  をセグメント毎に求め、セグメント毎にオーバーヘッド  $e$  を設定することによって、セグメントのサイズ（ひいては、コンテンツ C N のサイズ）に応じてオーバ

ッドeを最適化している。そのため、余分な冗長データが無くなり、コンテンツCNを配信する際のデータ量が低減し、マルチキャスト帯域（伝送帯域）を小さくできる。つまり、このサーバ1によれば、メディア分割を最適化し、メディア分割とメタコンテンツ化とを組み合わせたコンテンツのストリーミング配信を従来より小さな伝送帯域で実現できる。

#### 【0088】

以上、本発明に係る実施の形態について説明したが、本発明は上記実施の形態に限定されることなく様々な形態で実施される。

#### 【0089】

例えば、本実施の形態ではマルチキャスト型VODに適用したが、ユニキャスト型やブロードキャスト型等に適用可能であり、また、シネマコンプレックス等にも適用可能である。

#### 【0090】

また、本実施の形態ではメディア分割部及びFECエンコーダをソフトウェアで構成したが、ハードウェアで構成してもよい。

#### 【0091】

また、本実施の形態ではニュートン・ラフソン法を用いて任意ブロックにおけるオーバーヘッド及び再生時間を演算したが、二分割法等の他の数値解析法を用いて求めてもよい。

#### 【0092】

#### 【発明の効果】

本発明によれば、データサイズに応じてオーバーヘッドを最適化でき、メディア分割とメタコンテンツ化とを組み合わせたコンテンツの伝送におけるデータ量を低減することができる。

#### 【図面の簡単な説明】

#### 【図1】

本実施の形態に係るIPマルチキャスト型VODサービスの全体構成図である。

#### 【図2】

図 1 のサーバのメディア分割部における処理の説明図である。

【図 3】

同時受信チャネル数が 1 の場合のダウンロード時間と再生時間の関係を示す図である。

【図 4】

同時受信チャネル数が 2 の場合のダウンロード時間と再生時間の関係を示す図である。

【図 5】

同時受信チャネル数が 3 の場合のダウンロード時間と再生時間の関係を示す図である。

【図 6】

ダウンロード時間及び再生時間とストリーミングレートとの関係を示す図である。

【図 7】

図 1 のサーバのメディア分割部におけるブロック分割の説明図である。

【図 8】

ルビー変換におけるデータサイズとオーバーヘッドとの関係を示す図である。

【図 9】

オーバーヘッドを最適化した場合のダウンロード時間及び再生時間とストリーミングレートとの関係を示す図である。

【図 1 0】

ニュートン・ラフソン法の説明図ある。

【図 1 1】

図 1 のサーバの F E C エンコーダにおける処理の説明図である。

【図 1 2】

メタコンテンツ化の説明図である。

【図 1 3】

本実施の形態に係る再生時間及びダウンロード時間の演算処理を示すフローチャートである。

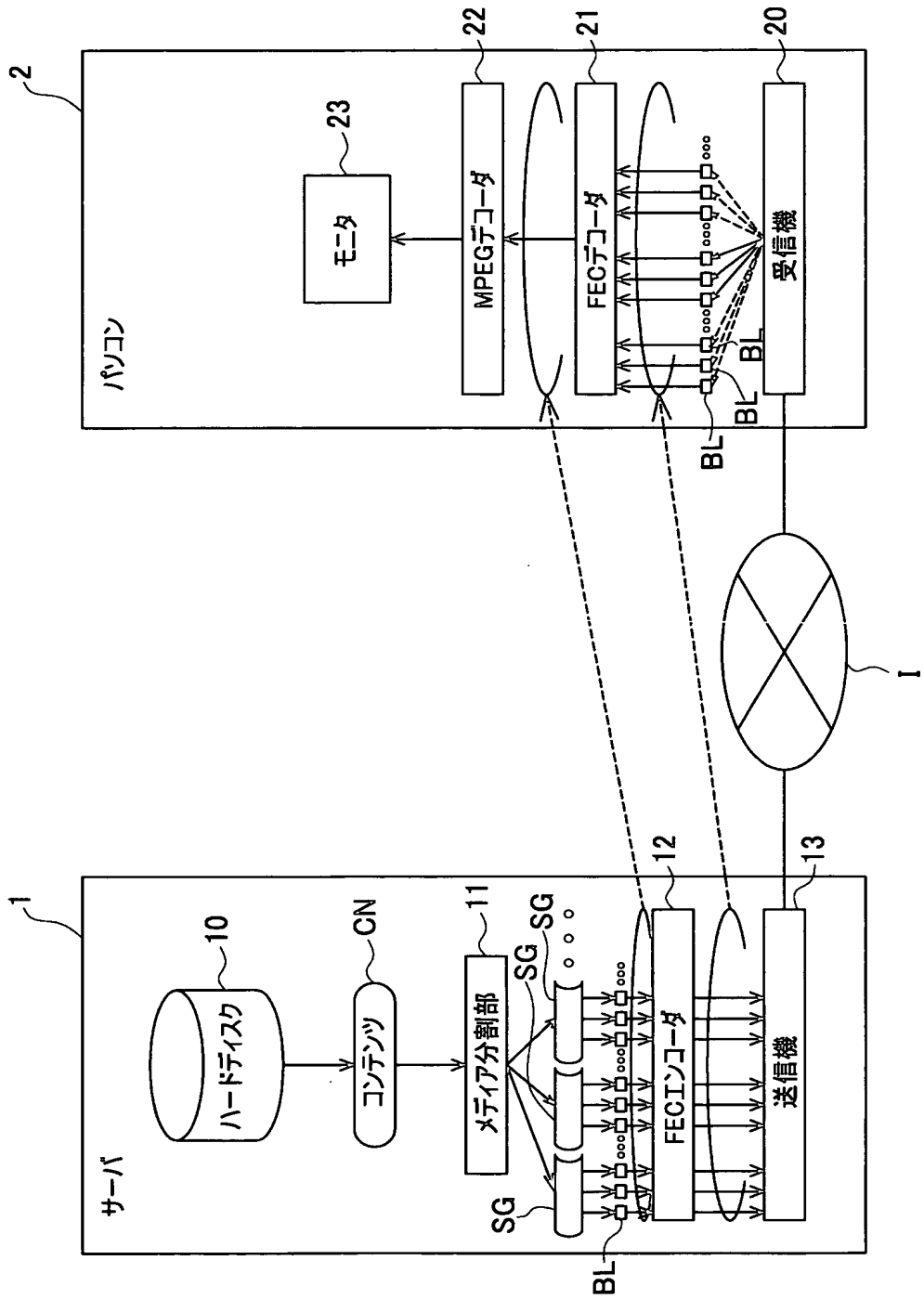
**【図 1 4】**

パケットロス耐性とマルチキャスト帯域との関係を示す図である。

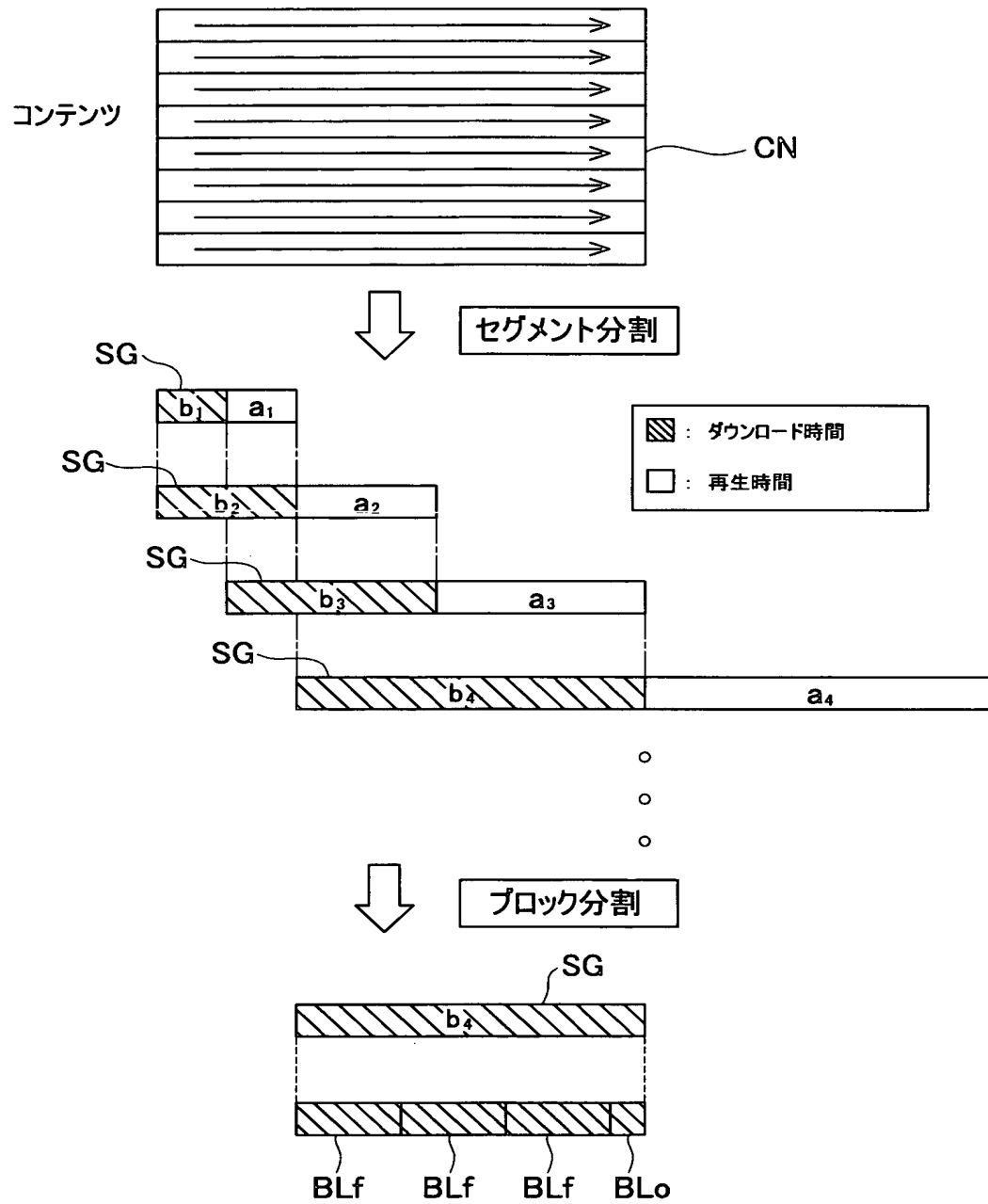
**【符号の説明】**

1…サーバ、2…パソコン、10…ハードディスク、11…メディア分割部、  
12…FECエンコーダ、13…送信機、20…受信機、21…FECデコーダ  
、22…MPEGデコーダ、23…モニタ

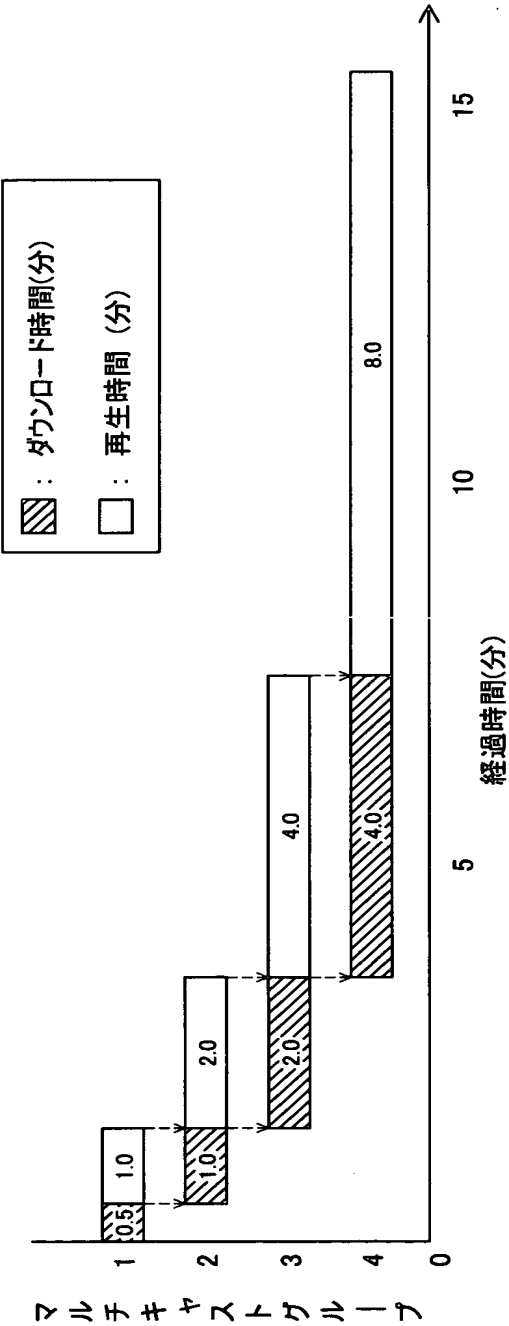
【書類名】 図面  
【図 1】



【図 2】

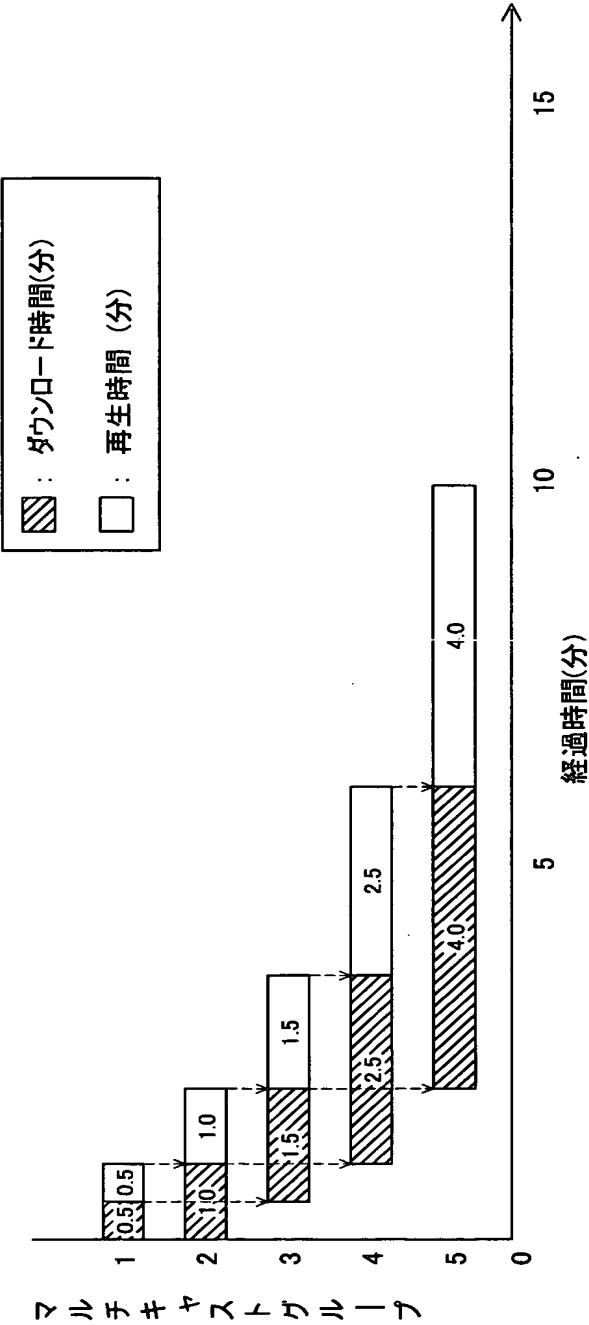


【図 3】

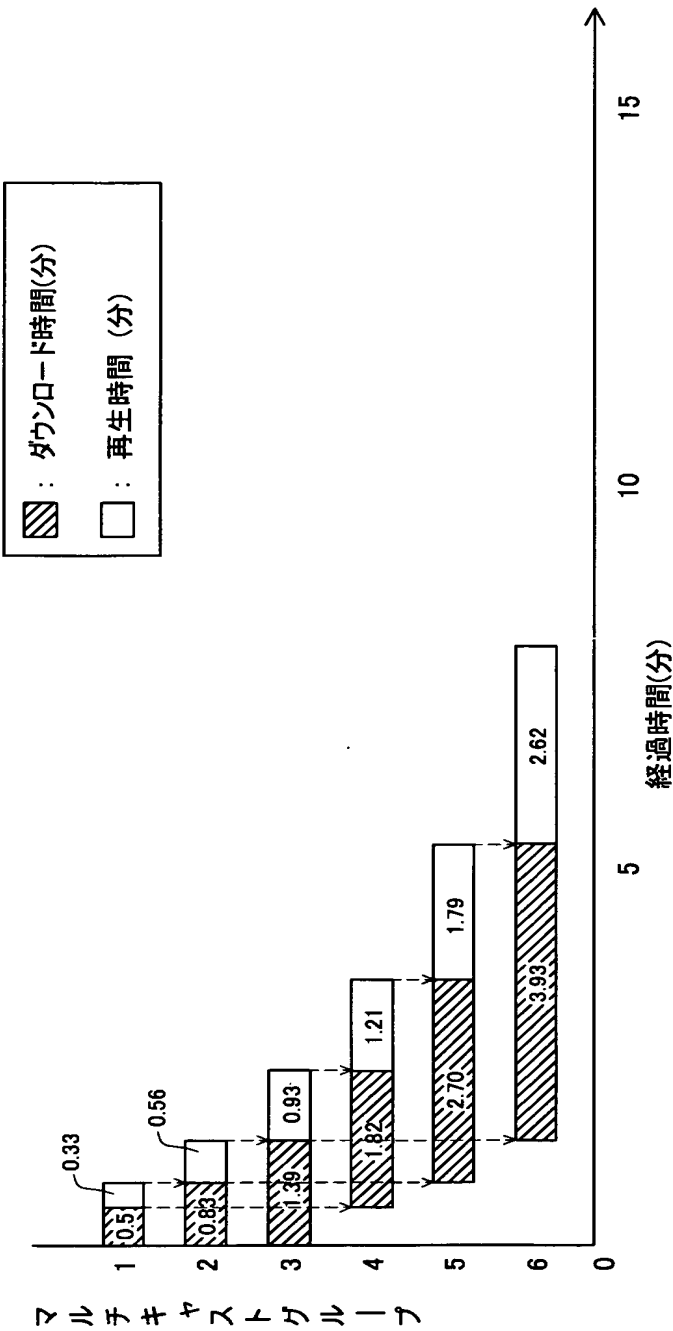




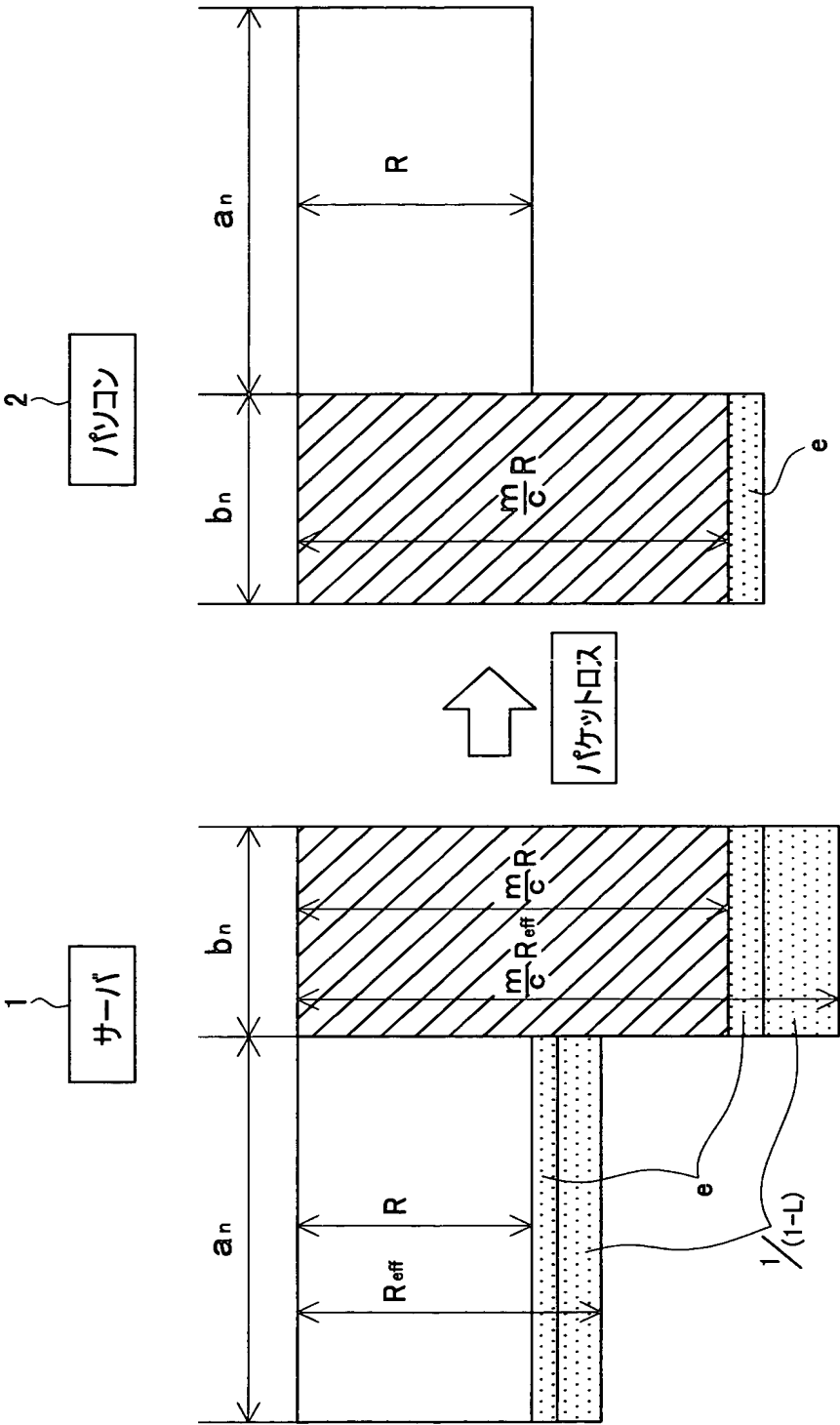
【図 4】



【図 5】

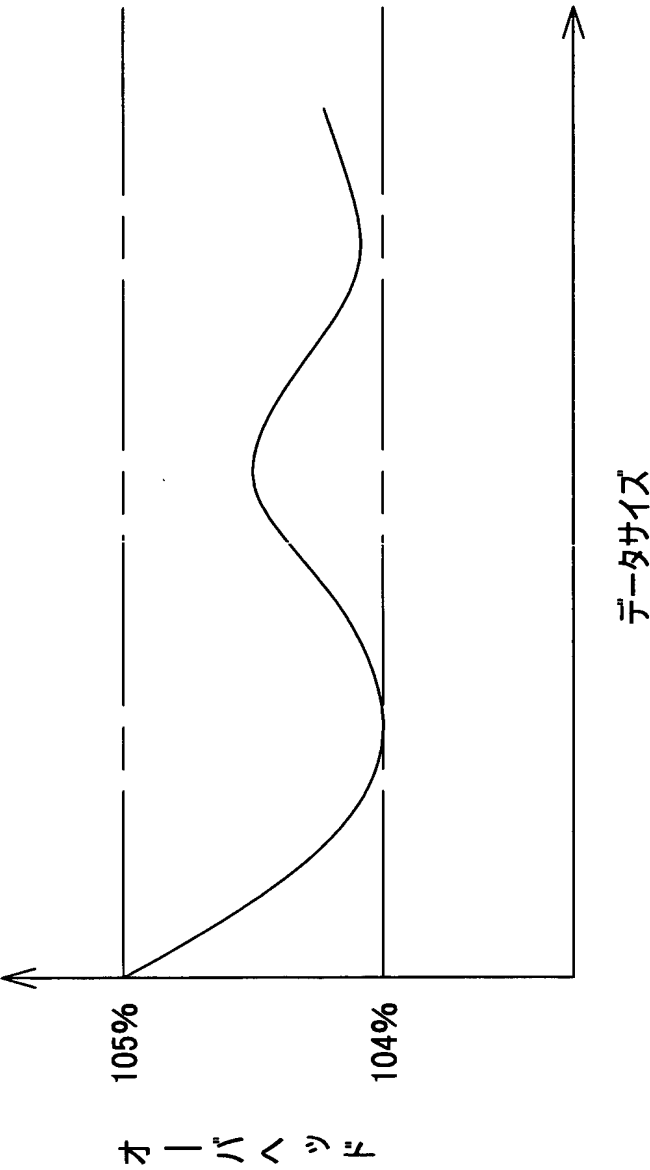


【図 6】

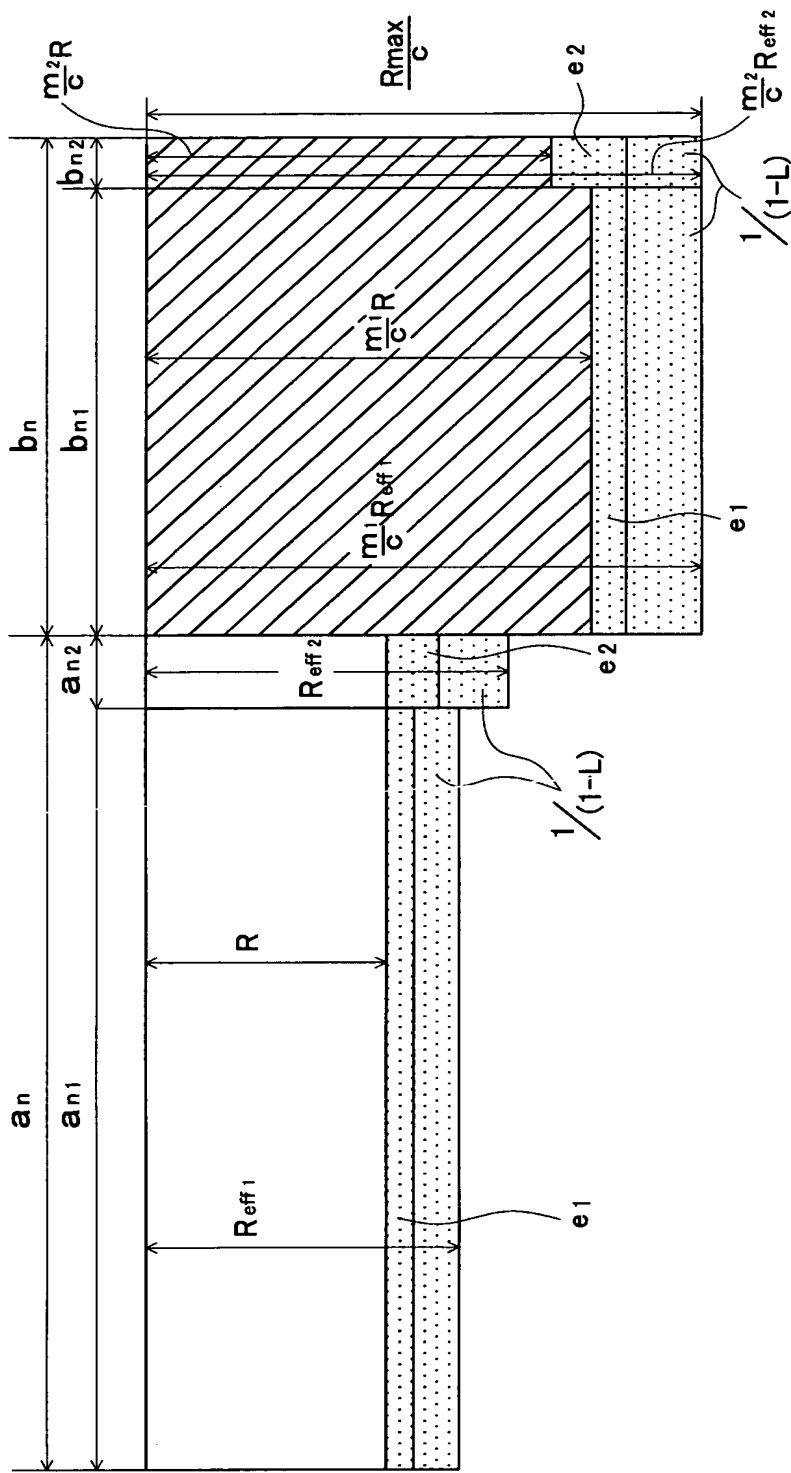




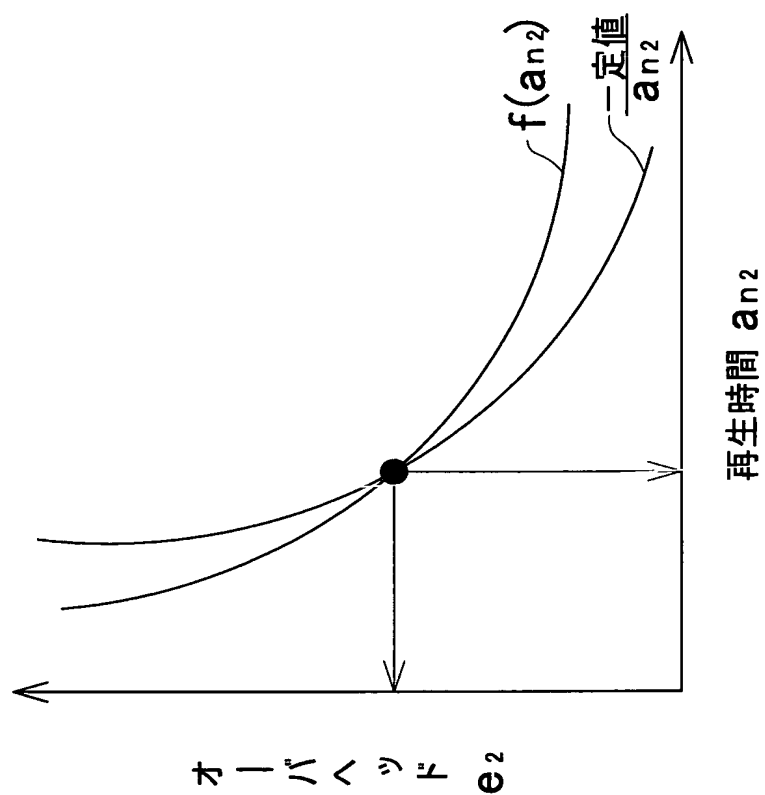
【図 8】



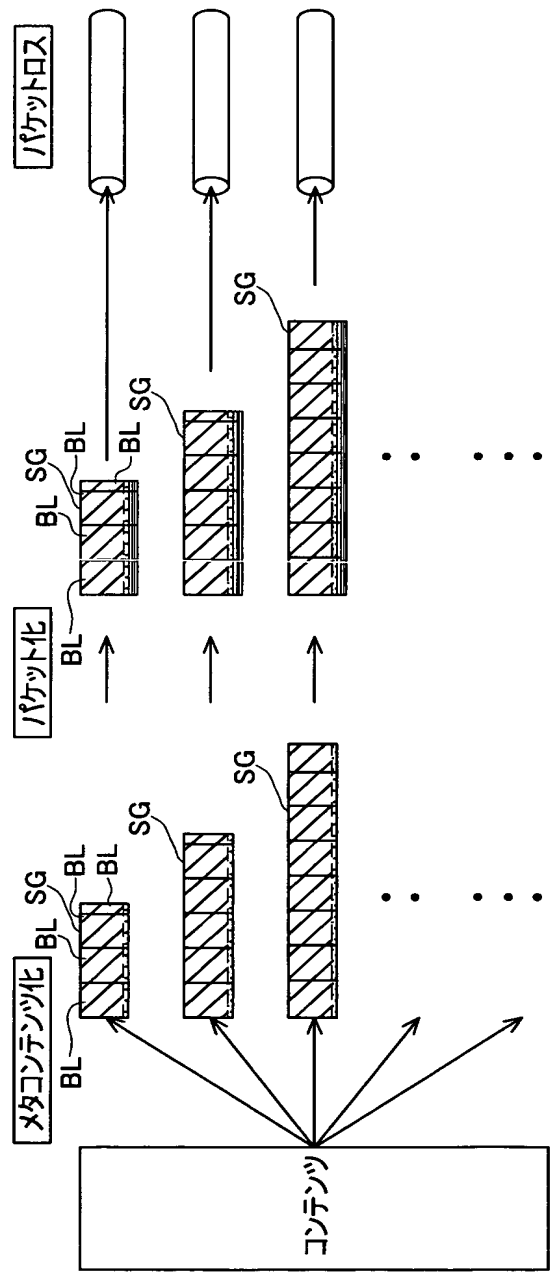
【図 9】



【図 10】

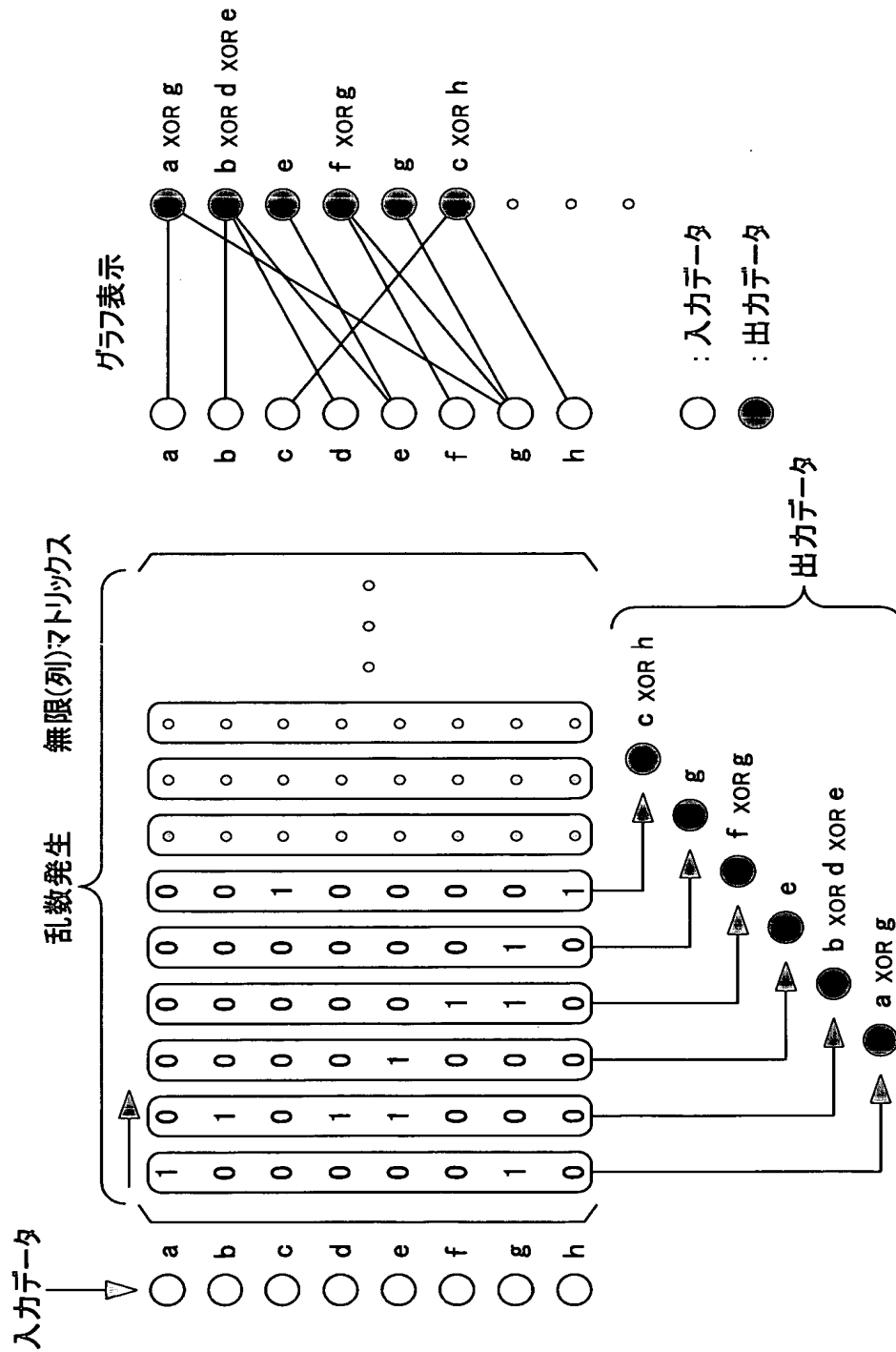


【図 11】

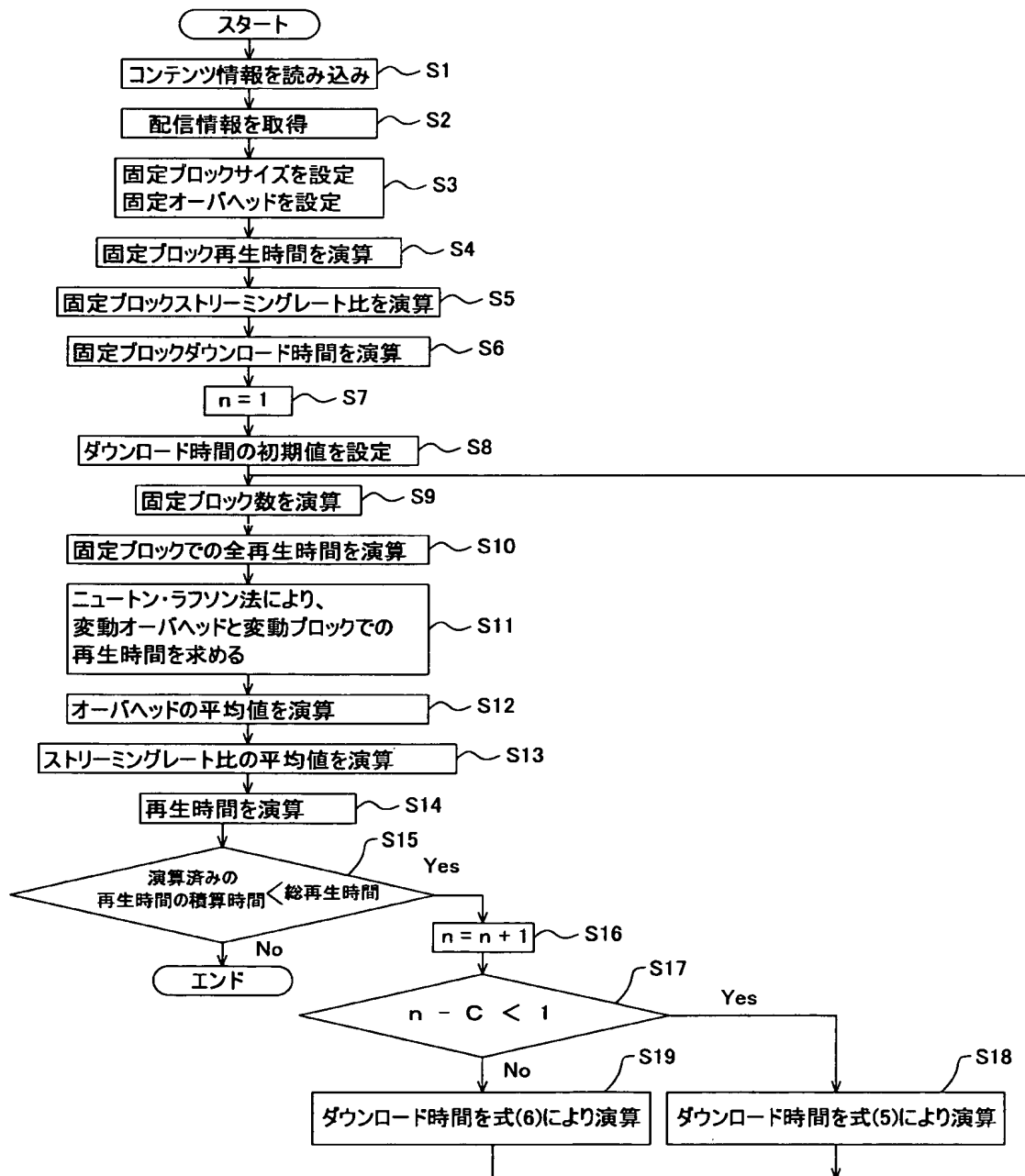




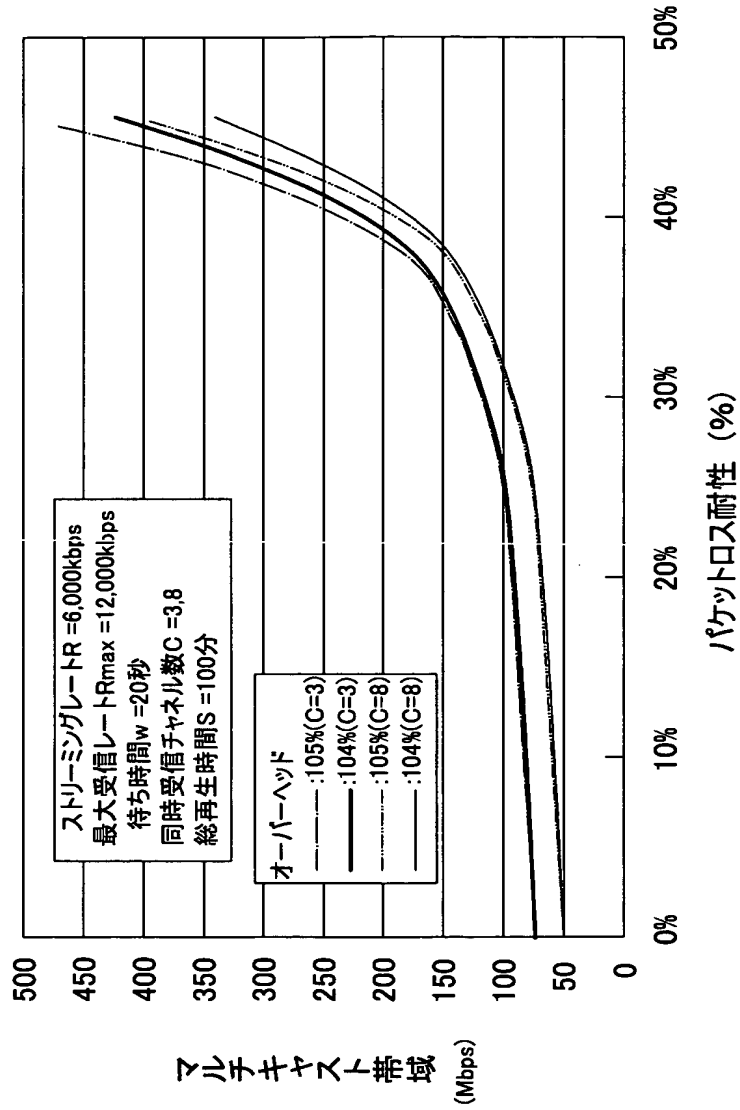
【図 12】



【図 13】



【図 14】



【書類名】 要約書

【要約】

【課題】 データサイズに応じてオーバーヘッドを最適化できる伝送データ生成方法及び伝送データ生成装置を提供することを課題とする。

【解決手段】 固定ブロックのサイズをオーバーヘッドに基づいて設定する工程（S3）と、固定ブロックのサイズに基づいて固定ブロックの再生時間を算出する工程（S4）と、コンテンツのセグメント毎に、固定ブロックの再生時間に基づいてセグメントの再生時間を算出する工程（S14）と、コンテンツのセグメント毎に、算出済みのセグメントの再生時間に基づいてセグメントの伝送時間を算出する工程（S18，S19）と、セグメントの伝送時間に応じてコンテンツをセグメントに分割する工程と、セグメントをブロックに分割する工程と、ブロック毎にコンテンツをオーバーヘッドを加味してメタコンテンツ化する工程とを含み、固定ブロックにおけるオーバーヘッドに基づいてセグメント毎にオーバーヘッドが設定されることを特徴とする。

【選択図】 図13

特願 2 0 0 3 - 1 3 2 2 9 2

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 2 1 3 0 ]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

大阪府大阪市中央区北浜四丁目 5 番 3 3 号

氏 名

住友電気工業株式会社